

明 細 書

認証システム及び遠隔分散保存システム

技術分野

本発明は、認証に関する情報の漏洩に強い認証システムとこの認証システムを用いて安全にデータを保存することができる遠隔分散保存システムに関する。

本願は、2003年10月28日に出願された特願2003-367527号に対し優先権を主張し、その内容をここに援用する。

背景技術

従来からユーザの端末装置とサーバとの間において認証を行う方法として、ユーザIDとユーザのみが知っているパスワードとを端末装置から入力し、サーバ側に蓄えられている情報と一致すれば、正当なユーザであることを認証する方法が知られている。

しかし、この方法は、端末装置とサーバ間の通信経路中において、不正にこれらの情報が盗まれてしまうと、簡単に不正利用を許してしまうため、SSL（文献1）、TLS（文献2）、SSH（文献3）等の暗号化技術を用いて、情報の送受信を行うのが一般的である。これは、パスワードと秘密の値と公開されている値とを使用して認証を行うものである。

（文献1）A. Frier, P. Karlton, and P. Kocher. The SSL 3.0 Protocol. Netscape Communications Corp., 1996, <http://wp.netscape.com/eng/ssl3/>

（文献2）IETF (Internet Engineering Task Force). Transport Layer security (tls) Charter. <http://www.ietf.org/html.charters/tls-charter.html>

（文献3）IETF (Internet Engineering Task Force). Secure Shell (secsh) Charter. <http://www.ietf.org/html.charters/secsh-charter.html>

BEST AVAILABLE COPY

しかしながら、非特許文献 1 ～ 3 に示す方法は、ユーザの端末装置側からパスワードで暗号化された情報あるいはサーバ側からパスワード認証データが漏れた場合は、オフラインの解析作業によってパスワードを求めることができってしまうという問題がある。サーバに対してオンラインでパスワード入力を繰り返し行う方法は、パスワードを間違えた回数に応じてアクセスを拒否するなどの対策を講じることが可能であるが、オフラインの解析作業は、防止策を講じることができないという問題もある。また、パスワードが漏洩してしまった場合、このパスワードでログインできるシステム内に保存されているデータも漏洩してしまうという問題もある。

発明の開示

本発明の目的は、情報の漏洩に強く、安全に暗号鍵の交換を行うことができる認証システムを提供することである。

本発明の目的は、本発明による認証システムを用いて安全にデータを保存することができる遠隔分散保存システムを提供することである。

本発明の要旨は、端末装置とサーバ間において相互に認証を行う認証システムであって、前記端末装置は、ユーザが予め決定しておいたパスワードに基づいて、サーバ登録用のパスワード認証データ H とユーザ保存用の認証情報 P' を求めるデータ伸長手段と、前記データ伸長手段によって求めた認証情報 P' を予め記憶しておく記憶手段と、前記記憶手段から読み出した認証情報 P' と認証時に入力されたパスワードを入力として所定の計算式により値 P を求める結合手段と、前記値 P と内部において発生させた乱数を入力として所定の計算式により値 Y_1 を求め、前記サーバへ送信するマスク演算手段と、前記値 P と内部において発生させた乱数と前記サーバから受信した値 Y_2 を入力として所定の計算式により値 MK を求めるマスター鍵生成手段と、前記値 MK を入力として所定の計算式により値 V_1 を求め、サーバへ送信するとともに、前記サーバから受信した値 V_2 と前記値 MK を入力として所定の計算式による値 V_2 と比較照合し、一致した場合にサーバを認証する認証結果判断手段とを備

え、前記サーバは、前記データ伸長手段によって求めたパスワード認証データHを予め記憶しておく記憶手段と、前記記憶手段から読み出したパスワード認証データHと内部において発生させた乱数を入力として所定の計算式により値Y2を求め、前記端末装置へ送信するマスク演算手段と、前記パスワード認証データHと内部において発生させた乱数と前記端末装置から受信した値Y1を入力として所定の計算式により値MKを求めるマスター鍵生成手段と、前記値MKを入力として所定の計算式により値V2を求め、端末装置へ送信するとともに、前記端末装置から受信した値V1と前記値MKを入力として所定の計算式による値V1と比較照合し、一致した場合に端末装置を認証する認証結果判断手段とを備える。

本発明の要旨は、端末装置とサーバ間において相互に認証を行う認証システムにおける端末装置上で動作する認証プログラムであって、ユーザが予め決定しておいたパスワードに基づいて、サーバ登録用のパスワード認証データHとユーザ保存用の認証情報P'を求めるデータ伸長処理と、前記データ伸長処理によって求めた認証情報P'を予め記憶しておく記憶処理と、前記記憶処理により記憶しておいた認証情報P'と認証時に入力されたパスワードを入力として所定の計算式により値Pを求める結合処理と、前記値Pと内部において発生させた乱数を入力として所定の計算式により値Y1を求め、前記サーバへ送信するマスク演算処理と、前記値Pと内部において発生させた乱数と前記サーバから受信した値Y2を入力として所定の計算式により値MKを求めるマスター鍵生成処理と、前記値MKを入力として所定の計算式により値V1を求め、サーバへ送信するとともに、前記サーバから受信した値V2と前記値MKを入力として所定の計算式による値V2と比較照合し、一致した場合にサーバを認証する認証結果判断処理とをコンピュータに行わせる。

本発明の要旨は、端末装置とサーバ間において相互に認証を行う認証システムにおけるサーバ上で動作する認証プログラムであって、パスワード認証データHを予め記憶しておく記憶処理と、前記記憶処理により記憶しておいたパス

ワード認証データHと内部において発生させた乱数を入力として所定の計算式により値Y2を求め、前記端末装置へ送信するマスク演算処理と、前記パスワード認証データHと内部において発生させた乱数と前記端末装置から受信した値Y1を入力として所定の計算式により値MKを求めるマスター鍵生成処理と、前記値MKを入力として所定の計算式により値V2を求め、端末装置へ送信するとともに、前記端末装置から受信した値V1と前記値MKを入力として所定の計算式による値V1と比較照合し、一致した場合に端末装置を認証する認証結果判断処理とをコンピュータに行わせる。

本発明の要旨は、端末装置とサーバ間において相互に認証を行う認証システムであって、前記端末装置は、ユーザが予め決定しておいたパスワードに基づいて、サーバ登録用のパスワード認証データHとユーザ保存用の認証情報P'を求めるデータ伸長手段と、前記データ伸長手段によって求めた認証情報P'とRSA鍵生成手段によって求めたRSA公開鍵(N, e)を予め記憶しておく記憶手段と、前記記憶手段から読み出した認証情報P'と認証時に入力されたパスワードを入力として所定の計算式により値Wを求める結合手段と、前記値Wと前記記憶手段から読み出したRSA公開鍵(N, e)と内部において発生させた乱数Tを入力として所定の計算式により値Zを求め、前記サーバへ送信するマスク演算手段と、前記サーバから受信した値V2と前記乱数Tを入力として所定の計算式による値V2と比較照合し、一致した場合にサーバを認証する認証結果判断手段と、前記乱数Tを入力として所定の計算式により値V1を求め、前記サーバへ送信する検証子生成手段とを備え、前記サーバは、RSA公開鍵(N, e)とRSA秘密鍵(N, d)を求めるRSA鍵生成手段と、前記RSA鍵生成手段によって求めたRSA秘密鍵(N, d)と前記データ伸長手段によって求めたパスワード認証データHを予め記憶しておく記憶手段と、前記記憶手段から読み出したRSA秘密鍵(N, d)とパスワード認証データHと前記端末装置から受信した値Zを入力として所定の計算式により値Tを求めるマスター鍵生成手段と、前記値Tを入力として所定の計算式により値V2を求め、前記端末装置へ送信する検証子生成手段と、前記端末装置から

受信した値 V_1 と前記値 T を入力として所定の計算式による値 V_1 と比較照合し、一致した場合に端末装置を認証する認証結果判断手段とを備える。

本発明の要旨は、端末装置とサーバ間において相互に認証を行う認証システムにおける端末装置上で動作する認証プログラムであって、ユーザが予め決定しておいたパスワードに基づいて、サーバ登録用のパスワード認証データ H とユーザ保存用の認証情報 P' を求めるデータ伸長処理と、前記データ伸長処理によって求めた認証情報 P' と $RS A$ 鍵生成処理によって求めた $RS A$ 公開鍵 (N, e) を予め記憶しておく記憶処理と、前記記憶処理により記憶しておいた認証情報 P' と認証時に入力されたパスワードを入力として所定の計算式により値 W を求める結合処理と、前記値 W と前記記憶処理により記憶しておいた $RS A$ 公開鍵 (N, e) と内部において発生させた乱数 T を入力として所定の計算式により値 Z を求め、前記サーバへ送信するマスク演算処理と、前記サーバから受信した値 V_2 と前記乱数 T を入力として所定の計算式による値 V_2 と比較照合し、一致した場合にサーバを認証する認証結果判断処理と、前記乱数 T を入力として所定の計算式により値 V_1 を求め、前記サーバへ送信する検証子生成処理とをコンピュータに行わせる。

本発明の要旨は、端末装置とサーバ間において相互に認証を行う認証システムにおけるサーバ上で動作する認証プログラムであって、 $RS A$ 公開鍵 (N, e) と $RS A$ 秘密鍵 (N, d) を求める $RS A$ 鍵生成処理と、前記 $RS A$ 鍵生成処理によって求めた $RS A$ 秘密鍵 (N, d) とパスワード認証データ H を予め記憶しておく記憶処理と、前記記憶処理により記憶しておいた $RS A$ 秘密鍵 (N, d) とパスワード認証データ H と前記端末装置から受信した値 Z を入力として所定の計算式により値 T を求めるマスター鍵生成処理と、前記値 T を入力として所定の計算式により値 V_2 を求め、前記端末装置へ送信する検証子生成処理と、前記端末装置から受信した値 V_1 と前記値 T を入力として所定の計算式による値 V_1 と比較照合し、一致した場合に端末装置を認証する認証結果判断処理とをコンピュータに行わせる。

本発明の要旨は、端末装置と複数のサーバ間において相互に認証を行い、前記端末装置内の保存対象のデータを前記サーバ内に分散して保存する遠隔分散保存システムであって、前記端末装置は、ユーザが予め決定しておいたパスワードに基づいて、サーバ登録用のパスワード認証データHとユーザ保存用の認証情報P'を求めるデータ伸長手段と、前記データ伸長手段によって求めた認証情報P'を予め記憶しておく記憶手段と、前記記憶手段から読み出した認証情報P'と認証時に入力されたパスワードを入力として所定の計算式により値Pを求める結合手段と、前記値Pと内部において発生させた乱数を入力として所定の計算式により値Y1を求め、前記サーバへ送信するマスク演算手段と、前記値Pと内部において発生させた乱数と前記サーバから受信した値Y2を入力として所定の計算式により値MKを求めるマスター鍵生成手段と、前記値MKを入力として所定の計算式により値V1を求め、サーバへ送信するとともに、前記サーバから受信した値V2と値V1と比較照合し、一致した場合にサーバを認証する認証結果判断手段と、サーバの認証が行われた場合に、セッション鍵SKをサーバの数だけ生成するセッション鍵生成手段と、前記保存対象のデータを分割して、認証したサーバの数と同数の分割データを得るデータ分割手段と、前記分割データのそれぞれと保存対象のデータを識別する識別情報とを、保存先のサーバと共有した前記セッション鍵SKを用いて暗号化して、各サーバに対して送信するデータ保存手段と、分割データが保存された各サーバから分割データを受信し、前記保存対象のデータを復元するデータ復元手段とを備え、前記サーバは、前記データ伸長手段によって求めたパスワード認証データHを予め記憶しておく記憶手段と、前記記憶手段から読み出したパスワード認証データHと内部において発生させた乱数を入力として所定の計算式により値Y2を求め、前記端末装置へ送信するマスク演算手段と、前記パスワード認証データHと内部において発生させた乱数と前記端末装置から受信した値Y1を入力として所定の計算式により値MKを求めるマスター鍵生成手段と、前記値MKを入力として所定の計算式により値V2を求め、端末装置へ送信するとともに、前記端末装置から受信した値V1と値V2と比較照合し、

一致した場合に端末装置を認証する認証結果判断手段と、端末装置の認証が行われた場合に、セッション鍵を生成するセッション鍵生成手段と、端末装置から受信した分割データを受信するデータ受信手段と、前記分割データを記憶するデータ記憶手段と、前記データ記憶手段に保存されている分割データを読み出して端末装置へ送信するデータ送信手段とを備えたことを特徴とする遠隔分散保存システム。

本発明の要旨は、端末装置と複数のサーバ間において相互に認証を行い、前記端末装置内の保存対象のデータを前記サーバ内に分散して保存する遠隔分散保存システムにおける端末装置上で動作する遠隔分散保存プログラムであって、ユーザが予め決定しておいたパスワードに基づいて、サーバ登録用のパスワード認証データHとユーザ保存用の認証情報P'を求めるデータ伸長処理と、前記データ伸長処理によって求めた認証情報P'を予め記憶しておく記憶処理と、前記記憶処理から読み出した認証情報P'と認証時に入力されたパスワードを入力として所定の計算式により値Pを求める結合処理と、前記値Pと内部において発生させた乱数を入力として所定の計算式により値Y1を求め、前記サーバへ送信するマスク演算処理と、前記値Pと内部において発生させた乱数と前記サーバから受信した値Y2を入力として所定の計算式により値MKを求めるマスター鍵生成処理と、前記値MKを入力として所定の計算式により値V1を求め、サーバへ送信するとともに、前記サーバから受信した値V2と値V1と比較照合し、一致した場合にサーバを認証する認証結果判断処理と、サーバの認証が行われた場合に、セッション鍵SKをサーバの数だけ生成するセッション鍵生成処理と、前記保存対象のデータを分割して、認証したサーバの数と同数の分割データを得るデータ分割処理と、前記分割データのそれぞれと保存対象のデータを識別する識別情報とを、保存先のサーバと共有した前記セッション鍵SKを用いて暗号化して、各サーバに対して送信するデータ保存処理と、分割データが保存された各サーバから分割データを受信し、前記保存対象のデータを復元するデータ復元処理とをコンピュータに行わせることを特徴とする遠隔分散保存プログラム

本発明の要旨は、端末装置と複数のサーバ間において相互に認証を行い、前記端末装置内の保存対象のデータを前記サーバ内に分散して保存する遠隔分散保存システムにおけるサーバ上で動作する遠隔分散保存プログラムであって、データ伸長処理によって求めたパスワード認証データHを予め記憶しておく記憶処理と、前記記憶処理から読み出したパスワード認証データHと内部において発生させた乱数を入力として所定の計算式により値Y2を求め、前記端末装置へ送信するマスク演算処理と、前記パスワード認証データHと内部において発生させた乱数と前記端末装置から受信した値Y1を入力として所定の計算式により値MKを求めるマスター鍵生成処理と、前記値MKを入力として所定の計算式により値V2を求め、端末装置へ送信するとともに、前記端末装置から受信した値V1と値V2と比較照合し、一致した場合に端末装置を認証する認証結果判断処理と、端末装置の認証が行われた場合に、セッション鍵を生成するセッション鍵生成処理と、端末装置から受信した分割データを受信するデータ受信処理と、前記分割データを記憶するデータ記憶処理と、前記データ記憶処理に保存されている分割データを読み出して端末装置へ送信するデータ送信処理とをコンピュータに行わせることを特徴とする遠隔分散保存プログラム。

図面の簡単な説明

図1は、本発明の一実施形態における端末装置の構成を示すブロック図である。

図2は、図1に示すデータ伸長器11の構成を示すブロック図である。

図3は、図1に示すデータ伸長器11の構成を示すブロック図である。

図4は、図1に示すデータ伸長器11の構成を示すブロック図である。

図5は、相互認証及び鍵交換を行う装置の構成を示すブロック図である。

図6は、図5に示す端末装置1の構成を示すブロック図である。

図7は、図5に示すサーバ2の構成を示すブロック図である。

図8は、図1に示すデータ伸長器11の構成を示すブロック図である。

図 9 は、パスワード認証データ更新－ 1 の端末装置 1 の構成を示すブロック図である。

図 10 は、パスワード認証データ更新－ 1 のサーバ 2 の構成を示すブロック図である。

図 11 は、パスワード認証データ更新－ 2 の端末装置 1 の構成を示すブロック図である。

図 12 は、パスワード認証データ更新－ 2 のサーバ 2 の構成を示すブロック図である。

図 13 は、図 1 に示すデータ伸長器 11 の構成を示すブロック図である。

図 14 は、図 1 に示すデータ伸長器 11 の構成を示すブロック図である。

図 15 は、安全な通信を利用して初期化処理を行う場合のサーバ 2 の構成を示すブロック図である。

図 16 は、安全ではない通信を利用して初期化処理を行う場合の端末装置 1 及びサーバ 2 の構成を示すブロック図である。

図 17 は、図 5 に示す端末装置 1 の構成を示すブロック図である。

図 18 は、図 5 に示すサーバ 2 の構成を示すブロック図である。

図 19 は、マスター鍵を利用して更新化処理を行う場合の端末装置 1 の構成を示すブロック図である。

図 20 は、マスター鍵を利用して更新化処理を行う場合のサーバ 2 の構成を示すブロック図である。

図 21 は、端末に分散データを保存しない場合の遠隔分散保存装置 5 の構成を示すブロック図である。

図 22 は、図 21 に示すデータ分散器 51 の構成を示すブロック図である。

図 23 は、端末にデータを保存しない場合の遠隔分散保存装置 5 の構成を示すブロック図である。

図 24 は、図 23 に示すデータ復元器 54 の構成を示すブロック図である。

図 25 は、端末にも分散データを保存する場合の遠隔分散保存装置 5 の構成を示すブロック図である。

図 26 は、図 25 に示すデータ分割器 51 の構成を示すブロック図である。

図 2 7 は、端末にも分散データを保存する場合の遠隔分散保存装置 5 の構成を示すブロック図である。

図 2 8 は、図 2 7 に示すデータ復元器 5 4 の構成を示すブロック図である。

発明を実施するための最良の形態

<第 1 実施例>

以下、図面を参照しつつ、本発明の好適な実施例について説明する。ただし、本発明は以下の各実施例に限定されるものではなく、例えばこれら実施例の構成要素同士を適宜組み合わせてもよい。

この認証システムは、ユーザの端末装置とサーバの認証装置がお互いに相互認証しながら同じセッション鍵を確保するためのシステムである。

ここで、以下の説明において用いる記号について説明しておく。

p , q は素数であり、 $q \mid p-1$ という関係がある。 $q \mid p-1$ は、 q は $p-1$ を割りきることのできる値であることを意味する。また、 g , h は $\text{mod } p$ 上の位数 q の有限体 (群) $G = \{g^j \text{ mod } p : 0 \leq j < q\}$ の生成元である (楕円曲線上の群でも同じように構成できる)。ここで、“ $g^j \text{ mod } p$ ” は、法指数演算で、 g を j 乗した値を p で割った残り (R e m a i n d e r) という意味である。ここで、 g は $(1 < g < p-1, g^q = 1 \text{ mod } p, g^j \neq 1 \text{ mod } p (0 < j < q))$ であり、 h は $h = g^a \text{ mod } p$ である。つまり、 p , q は演算体系 (素体の標数) を示す。例えば、 $H = h^x \text{ mod } p (0 < x < q)$ で x は秘密情報である (つまり、 H が与えられた時、 $x = \log_h H$ を求めるのは数学的に難しい問題; H の生成元 h に対する離散対数問題)。また、乱数発生器から発生される乱数は $R \in (Z/qZ)^*$ を無作為に生成する。ここで、 $(Z/qZ)^*$ は $\{1, 2, \dots, q\}$ の集合を示す。また、 N はパスワードの長さを示す。また、 \parallel は値を連結 (c o n c a t e n a t i o n) するという意味である。

<端末装置の初期化>

ユーザは、サーバに対して個人登録したい時、自分の端末装置の初期化を行

う。図1は、ユーザの端末装置の初期化処理の構成を示すブロック図である。初期化は、ユーザがパスワードを入力すると、データ伸張器11によって、サーバ登録用のパスワード認証データHと、ユーザ保存用の値P'が生成され、パスワード認証データHは、サーバに受け渡され、値P'は、メモリ12へ保存する。ここで、データ伸張器11は、多項式、多項式とハッシュ関数、ハッシュ関数、擬似乱数発生器などで構成することが可能である。

(1) 多項式を利用する場合 (その1)

初めに、図2を参照して、多項式を利用する場合 (その1) について説明する。

まず、多項式発生器111によりランダムに多項式を発生する。このとき、登録するサーバの数が一つだったらxを変数とする1次多項式 ($p'(x) = \alpha_1 \cdot x \bmod q$) を、サーバの数がn個だったらn次多項式 ($p'(x) = \alpha_1 \cdot x + \alpha_2 \cdot x^2 + \dots + \alpha_n \cdot x^n \bmod q$) を発生する。ここで、 α は $(\mathbb{Z}/q\mathbb{Z})^*$ から無作為に選ばれる。例えば、一つのサーバの場合、 $p'(x)$ は、 $p'(x) = \alpha_1 \cdot x \bmod q$ となる。ここで、ユーザは自分が覚えているパスワード (例えば、"PooH93") を入力する。多項式とユーザのパスワードが入力されたらパスワード認証データ生成器112は、パスワード認証データHを生成する。パスワード認証データHは、例えば $H = h^{p'(1) + \text{PooH}93} \bmod p$ により計算できる。ここで、 $p'(1)$ は $p'(x)$ でxの代わりにサーバのID (例えば、「1」) を入れて計算した値である。パスワード認証データHはユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ12は多項式発生器から発生された多項式 $P' = p'(x)$ を記憶して保存する。

(2) 多項式を利用する場合 (その2)

次に、図2を参照して、多項式を利用する場合 (その2) について説明する。

まず、多項式発生器111によりランダムに多項式を発生する。このとき、

登録するサーバの数が一つだったら x を変数とする 1 次多項式 ($p'(x) = \alpha_1 \cdot x \bmod q$) を、サーバの数が n 個だったら n 次多項式 ($p'(x) = \alpha_1 \cdot x + \alpha_2 \cdot x^2 + \dots + \alpha_n \cdot x^n \bmod q$) を発生する。ここで、 α は $(\mathbb{Z}/q\mathbb{Z})^*$ から無作為に選ばれる。例えば、一つのサーバの場合、 $p'(x)$ は、 $p'(x) = \alpha_1 \cdot x \bmod q$ となる。ここで、ユーザは自分が覚えているパスワード (例えば、"P o o h 9 3") を入力する。多項式とユーザのパスワードが入力されたらパスワード認証データ生成器 1 1 2 は、パスワード認証データ H を生成する。パスワード認証データ H は、例えば $H = p(1) = p'(1) + \text{P o o h 9 3} \bmod q$ により計算できる。ここで、 $p'(1)$ は $p'(x)$ で x の代わりにサーバの ID (例えば、「1」) を入れて計算した値である。パスワード認証データ H はユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ 1 2 は多項式発生器から発生された多項式 $P' = p'(x)$ を記憶して保存する。

(3) 多項式とハッシュ関数を利用する場合 (その 1)

次に、図 8 を参照して、多項式とハッシュ関数を利用する場合 (その 1) について説明する。

まず、多項式発生器 1 1 9 によりランダムに多項式を発生する。このとき、登録するサーバの数が一つだったら x を変数とする 1 次多項式 ($p'(x) = \alpha_1 \cdot x \bmod N$) を、サーバの数が n 個だったら n 次多項式 ($p'(x) = \alpha_1 \cdot x + \alpha_2 \cdot x^2 + \dots + \alpha_n \cdot x^n \bmod N$) を発生する。ここで、 α は $(\mathbb{Z}/q\mathbb{Z})^*$ から無作為に選ばれる。例えば、一つのサーバの場合、 $p'(x)$ は、 $p'(x) = \alpha_1 \cdot x \bmod N$ となる。そして、ハッシュ関数発生器 1 2 0 によりランダムにハッシュ関数 $HASH$ を発生する。 $HASH$ は一方向ハッシュ関数である。ここで、ユーザは自分が覚えているパスワード (例えば、"P o o h 9 3") を入力する。多項式とハッシュ関数とユーザのパスワードが入力されたらパスワード認証データ生成器 1 2 1 は、パスワード認証データ H を生成する。パスワード認証データ H は、例えば $H = h^{p'(1)} \bmod p$ により計

算できる。ここで $p(1)$ は、 $p(1) = p'(1) + \text{HASH}(\text{P o o h 9 3} \parallel \text{ID}(U) \parallel \text{ID}(S)) \bmod N$ により計算する。ここで、 $\text{ID}(U)$ と $\text{ID}(S)$ はそれぞれユーザとサーバの ID を表す。ここで、 $p'(1)$ は $p'(x)$ で x の代わりに「1」を入れて計算した値である。

例えば、登録するサーバの数が n の場合、パスワード認証データ生成器 121 は、 i 番目のサーバに対してパスワード認証データ H を生成する。パスワード認証データ H は、例えば $H = h^{p(i)} \bmod p$ により計算できる。ここで $p(i)$ は、 $p(i) = p'(i) + \text{HASH}(\text{P o o h 9 3} \parallel \text{ID}(U) \parallel \text{ID}(S)) \bmod N$ により計算する。ここで、 $\text{ID}(U)$ と $\text{ID}(S)$ はそれぞれユーザと i 番目のサーバの ID を表す。ここで、 $p'(i)$ は n 次多項式 $p'(x)$ で x の代わりに「 i 」を入れて計算した値である。

パスワード認証データ H はユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ 12 は多項式発生器から発生された多項式 $p'(x)$ とハッシュ関数発生器から発生されたハッシュ関数 HASH とを一緒に $P' = (p'(x), \text{HASH})$ として記憶して保存する。

(4) 多項式とハッシュ関数を利用する場合 (その2)

次に、図 8 を参照して、多項式とハッシュ関数を利用する場合 (その2) について説明する。

まず、多項式発生器 119 によりランダムに多項式を発生する。このとき、登録するサーバの数が一つだったら x を変数とする 1 次多項式 ($p'(x) = \alpha_1 \cdot x \bmod N$) を、サーバの数が n 個だったら n 次多項式 ($p'(x) = \alpha_1 \cdot x + \alpha_2 \cdot x^2 + \dots + \alpha_n \cdot x^n \bmod N$) を発生する。ここで、 α は $(\mathbb{Z}/q\mathbb{Z})^*$ から無作為に選ばれる。例えば、一つのサーバの場合、 $p'(x)$ は、 $p'(x) = \alpha_1 \cdot x \bmod N$ となる。そして、ハッシュ関数発生器 120 によりランダムにハッシュ関数 HASH を発生する。 HASH は一方向ハッシュ関数である。ここで、ユーザは自分が覚えているパスワード (例えば、「P o o h 9 3」) を入力する。多項式とハッシュ関数とユーザのパスワードが

入力されたらパスワード認証データ生成器 1 2 1 は、パスワード認証データ H を生成する。パスワード認証データ H は、例えば $H = p(1) = p'(1) + \text{HASH}(\text{P o o h 9 3} \parallel \text{ID}(U) \parallel \text{ID}(S)) \bmod N$ により計算できる。ここで、 $\text{ID}(U)$ と $\text{ID}(S)$ はそれぞれユーザとサーバの ID を表す。ここで、 $p'(1)$ は $p'(x)$ で x の代わりに「1」を入れて計算した値である。

例えば、登録するサーバの数が n の場合、パスワード認証データ生成器 1 2 1 は、 i 番目のサーバに対してパスワード認証データ H を生成する。パスワード認証データ H は、例えば $H = p(i) = p'(i) + \text{HASH}(\text{P o o h 9 3} \parallel \text{ID}(U) \parallel \text{ID}(S)) \bmod N$ により計算できる。ここで、 $\text{ID}(U)$ と $\text{ID}(S)$ はそれぞれユーザと i 番目のサーバの ID を表す。ここで、 $p'(i)$ は n 次多項式 $p'(x)$ で x の代わりに「 i 」を入れて計算した値である。

パスワード認証データ H はユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ 1 2 は多項式発生器から発生された多項式 $p'(x)$ とハッシュ関数発生器から発生されたハッシュ関数 HASH とを一緒に $P' = (p'(x), \text{HASH})$ として記憶して保存する。

(5) ハッシュ関数を利用する場合 (その 1)

次に、図 3 を参照して、ハッシュ関数を利用する場合 (その 1) について説明する。

まず、ハッシュ関数発生器 1 1 3 によりランダムにハッシュ関数 HASH を発生する。そして、秘密値発生器 1 1 4 もランダムに秘密値 S を発生する。ここで、ユーザは自分が覚えているパスワード (例えば、「P o o h 9 3」) を入力する。ハッシュ関数 HASH と秘密値 S とユーザのパスワードが入力されたらパスワード認証データ生成器 1 1 5 はパスワード認証データ H を生成する。パスワード認証データ H は、例えば、 $H = h^{\text{HASH}(S \parallel \text{P o o h 9 3} \parallel \text{ID}(U) \parallel \text{ID}(S))} \bmod p$ により計算できる。ここで、 $\text{ID}(U)$ と $\text{ID}(S)$ はそれぞれユー

ザとサーバのIDを表す。パスワード認証データHはユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ12はハッシュ関数発生器113と秘密値発生器114から発生されたハッシュ関数HASHと秘密値Sとを一緒に $P' = (S, HASH)$ として記憶して保存する。

(6) ハッシュ関数を利用する場合 (その2)

次に、図3を参照して、ハッシュ関数を利用する場合 (その2) について説明する。

まず、ハッシュ関数発生器113によりランダムにハッシュ関数HASHを発生する。そして、秘密値発生器114もランダムに秘密値Sを発生する。ここで、ユーザは自分が覚えているパスワード (例えば、"P o o h 9 3") を入力する。ハッシュ関数HASHと秘密値Sとユーザのパスワードが入力されたらパスワード認証データ生成器115はパスワード認証データHを生成する。パスワード認証データHは、例えば、 $H = HASH(S \parallel P o o h 9 3 \parallel ID(U) \parallel ID(S)) \bmod q$ により計算できる。ここで、ID(U)とID(S)はそれぞれユーザとサーバのIDを表す。パスワード認証データHはユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ12はハッシュ関数発生器113と秘密値発生器114から発生されたハッシュ関数HASHと秘密値Sとを一緒に $P' = (S, HASH)$ として記憶して保存する。

(7) 擬似乱数発生器を利用する場合 (その1)

次に、図4を参照して、擬似乱数発生器を利用する場合 (その1) について説明する。

まず、擬似乱数発生器116によりランダムに擬似乱数関数PRNGを発生する。そして、秘密値発生器117もランダムに秘密値Sを発生する。ここで、ユーザは自分が覚えているパスワード (例えば、"P o o h 9 3") を入力す

る。擬似乱数関数 P R N G と秘密値 S とユーザのパスワードが入力されたのを受けて、パスワード認証データ生成器 1 1 8 はパスワード認証データ H を生成する。パスワード認証データ H は、例えば、 $H = h^{PRNG(S \parallel P o o h 9 3 \parallel ID(U) \parallel ID(S))} \bmod p$ により計算できる。ここで、ID (U) と ID (S) はそれぞれユーザとサーバの ID を表す。パスワード認証データ H はユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ 1 2 は擬似乱数発生器 1 1 6 と秘密値発生器 1 1 7 から発生させた擬似乱数関数 P R N G と秘密値 S とを一緒に $P' = (S, P R N G)$ として記憶して保存する。

(8) 擬似乱数発生器を利用する場合 (その 2)

次に、図 4 を参照して、擬似乱数発生器を利用する場合 (その 2) について説明する。

まず、擬似乱数発生器 1 1 6 によりランダムに擬似乱数関数 P R N G を発生する。そして、秘密値発生器 1 1 7 もランダムに秘密値 S を発生する。ここで、ユーザは自分が覚えているパスワード (例えば、" P o o h 9 3 ") を入力する。擬似乱数関数 P R N G と秘密値 S とユーザのパスワードが入力されたのを受けて、パスワード認証データ生成器 1 1 8 はパスワード認証データ H を生成する。パスワード認証データ H は、例えば、 $H = PRNG(S \parallel P o o h 9 3 \parallel ID(U) \parallel ID(S)) \bmod q$ により計算できる。ここで、ID (U) と ID (S) はそれぞれユーザとサーバの ID を表す。パスワード認証データ H はユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ 1 2 は擬似乱数発生器 1 1 6 と秘密値発生器 1 1 7 から発生させた擬似乱数関数 P R N G と秘密値 S とを一緒に $P' = (S, P R N G)$ として記憶して保存する。

次に、図 6、7 を参照して、前述した初期化を行った端末装置 1 とサーバ 2 (図 5 参照) との間で相互認証及び鍵交換を行う動作を説明する。

＜端末装置の動作＞

(1) 多項式を利用した場合（その1とその2）

初めに、多項式を利用した場合の端末装置1の動作を説明する。端末装置1は、前述した多項式を利用した場合（その1）、多項式を利用した場合（その2）に関わらず次のように動作する。

まず、ユーザの端末装置1に備えたメモリ12から記憶された多項式 $P' = p'(x)$ を読み出す。結合器32はメモリ12から読み出した多項式 P' とユーザが入力したパスワードにより $P = p(x)$ を計算して出力する。例えば、 $p(x) = p'(x) + P o o h 9 3 = \alpha_1 \cdot x + P o o h 9 3 \bmod q$ により計算する。マスク演算器34は、結合器32から入力された P と乱数発生器33において発生させた乱数 R_1 とから Y_1 を、 $Y_1 = g^{R_1} \cdot h^{-p(1)} \bmod p$ により計算する。ここで $p(1)$ は、 $p(1) = p'(1) + P o o h 9 3 = \alpha_1 \cdot 1 + P o o h 9 3 \bmod q$ により計算する。ここで、「1」はサーバの認証IDを表す。通信処理部35は Y_1 をサーバ2へ送信し、サーバ2から Y_2 を受信する。マスター鍵生成器36は結合器32から出力される P と乱数発生器33から出力される R_1 と受信した Y_2 を入力として MK を、 $MK = (Y_2 \cdot h^{-p(1)})^{R_1} \bmod p$ により計算して出力する。

続いて認証結果判断部37は、 MK を入力として、 $V_1 = \text{HASH}(00 \parallel Y_1 \parallel Y_2 \parallel MK)$ により V_1 を計算してこの V_1 を通信処理部35によりサーバ2へ送信し、サーバ2から受信した V_2 と $\text{HASH}(01 \parallel Y_1 \parallel Y_2 \parallel MK)$ を比較する。ここで、 HASH は一方向ハッシュ関数であるし、 HASH の代わりに MAC (Message Authentication Code) を使ってもよい。

次に、認証結果判断部37において V_2 と $\text{HASH}(01 \parallel Y_1 \parallel Y_2 \parallel MK)$ が一致しない場合、認証結果判断部37は、エラー発生器38に対して、一致しないことを通知する。これを受けて、エラー発生器38はエラーを発生して処理を中断する。一方、認証結果判断部37において V_2 と $\text{HASH}(01 \parallel Y_1 \parallel Y_2 \parallel MK)$ が一致した場合はサーバ2が正当な装置として認証してセッション鍵生成器39は、 $SK = \text{HASH}(11 \parallel Y_1 \parallel Y_2 \parallel MK)$ によりセッ

ション鍵SKを生成する。

(2) 多項式とハッシュ関数を利用した場合 (その1とその2)

次に、多項式とハッシュ関数を利用した場合の端末装置1の動作を説明する。端末装置1は、前述した多項式とハッシュ関数を利用した場合 (その1)、多項式とハッシュ関数を利用した場合 (その2) に関わらず次のように動作する。

まず、ユーザの端末装置1に備えたメモリ12から記憶された多項式とハッシュ関数 $P' = (p'(x), \text{HASH})$ を読み出す。結合器32はメモリ12から読み出した多項式 $p'(x)$ とハッシュ関数 HASH とユーザが入力したパスワードにより $P = p(x)$ を計算して出力する。例えば、 $p'(x)$ が1次多項式の場合、 $p(x) = p'(x) + \text{HASH}(\text{P o o h 9 3} \parallel \text{ID}(U) \parallel \text{ID}(S)) = \alpha_1 \cdot x + \text{HASH}(\text{P o o h 9 3} \parallel \text{ID}(U) \parallel \text{ID}(S)) \bmod N$ により計算する。マスク演算器34は、結合器32から入力された P と乱数発生器33において発生させた乱数 R_1 とから Y_1 を、 $Y_1 = g^{R_1} \cdot h^{-p(1)} \bmod p$ により計算する。ここで $p(1)$ は、 $p(1) = p'(1) + \text{HASH}(\text{P o o h 9 3} \parallel \text{ID}(U) \parallel \text{ID}(S)) = \alpha_1 \cdot 1 + \text{HASH}(\text{P o o h 9 3} \parallel \text{ID}(U) \parallel \text{ID}(S)) \bmod N$ により計算する。ここで、 $p'(1)$ は $p'(x)$ で x の代わりに「1」入れて計算した値である。通信処理部35は Y_1 をサーバ2へ送信し、サーバ2から Y_2 を受信する。マスター鍵生成器36は結合器32から出力される P と乱数発生器33から出力される R_1 と受信した Y_2 を入力として MK を、 $MK = (Y_2 \cdot h^{-p(1)})^{R_1} \bmod p$ により計算して出力する。

ユーザの端末装置1に備えたメモリ12から読み出した多項式 $p'(x)$ が n 次多項式の場合、結合器32は多項式 $p'(x)$ とハッシュ関数 HASH とユーザが入力したパスワードにより $P = p(x)$ を計算して出力する。例えば、 $p(x) = p'(x) + \text{HASH}(\text{P o o h 9 3} \parallel \text{ID}(U) \parallel \text{ID}(S)) \bmod N$ により計算する。マスク演算器34は、結合器32から入力された P と乱数発生器33において発生させた乱数 R_1 とから Y_1 を、 $Y_1 = g^{R_1} \cdot h^{-p(i)} \bmod p$ により計算する。ここで $p(i)$ は、 $p(i) = p'(i) + \text{HASH}$

$H(P o o h 9 3 \parallel I D(U) \parallel I D(S)) \bmod N$ により計算する。ここで、 $p'(i)$ は*i*番目のサーバに対して $p'(x)$ で*x*の代わりに「*i*」入れて計算した値である。通信処理部35は Y_1 をサーバ2へ送信し、サーバ2から Y_2 を受信する。マスター鍵生成器36は結合器32から出力される P と乱数発生器33から出力される R_1 と受信した Y_2 を入力として MK を、 $MK = (Y_2 \cdot h^{-p(i)})^{R_1} \bmod p$ により計算して出力する。

続いて認証結果判断部37は、 MK を入力として、 $V_1 = HASH(00 \parallel Y_1 \parallel Y_2 \parallel MK)$ により V_1 を計算してこの V_1 を通信処理部35によりサーバ2へ送信し、サーバ2から受信した V_2 と $HASH(01 \parallel Y_1 \parallel Y_2 \parallel MK)$ を比較する。ここで、 $HASH$ は一方向ハッシュ関数であるし、 $HASH$ の代わりに MAC (Message Authentication Code) を使ってもよい。

次に、認証結果判断部37において V_2 と $HASH(01 \parallel Y_1 \parallel Y_2 \parallel MK)$ が一致しない場合、認証結果判断部37は、エラー発生器38に対して、一致しないことを通知する。これを受けて、エラー発生器38はエラーを発生して処理を中断する。一方、認証結果判断部37において V_2 と $HASH(01 \parallel Y_1 \parallel Y_2 \parallel MK)$ が一致した場合はサーバ2が正当な装置として認証してセッション鍵生成器39は、 $SK = HASH(11 \parallel Y_1 \parallel Y_2 \parallel MK)$ によりセッション鍵 SK を生成する。

(3) ハッシュ関数を利用した場合 (その1とその2)

次に、ハッシュ関数を利用した場合の端末装置1の動作を説明する。端末装置1は、前述したハッシュ関数を利用した場合 (その1)、ハッシュ関数を利用した場合 (その2) に関わらず次のように動作する。

まず、ユーザの端末装置1に備えたメモリ12から記憶された秘密値とハッシュ関数 $P' = (S, HASH)$ を読み出す。結合器32はメモリ12から読み出した秘密値 S とハッシュ関数 $HASH$ とユーザが入力したパスワードにより $P = p$ を計算して出力する。例えば、 p は、 $p = HASH(S \parallel P o o h 9 3 \parallel I D(U) \parallel I D(S)) \bmod q$ により計算する。マスク演算器34

は、結合器 3 2 から入力された P と乱数発生器 3 3 において発生させた乱数 R_1 とから Y_1 を、 $Y_1 = g^{R_1} \cdot h^{-p} \bmod p$ により計算する。通信処理部 3 5 は、 Y_1 をサーバ 2 へ送信し、サーバ 2 から Y_2 を受信する。マスター鍵生成器 3 6 は結合器 3 2 から出力される P と乱数発生器 3 3 から出力される R_1 と受信した Y_2 を入力として MK を、 $MK = (Y_2 \cdot h^{-p})^{R_1} \bmod p$ により計算して出力する。

続いて認証結果判断部 3 7 は、 MK を入力として、 $V_1 = \text{HASH}(00 \parallel Y_1 \parallel Y_2 \parallel MK)$ により V_1 を計算してこの V_1 を通信処理部 3 5 によりサーバ 2 へ送信し、サーバ 2 から受信した V_2 と $\text{HASH}(01 \parallel Y_1 \parallel Y_2 \parallel MK)$ を比較する。ここで、 HASH は一方向ハッシュ関数であるし、 HASH の代わりに MAC (Message Authentication Code) を使ってもよい。

次に、認証結果判断部 3 7 において V_2 と $\text{HASH}(01 \parallel Y_1 \parallel Y_2 \parallel MK)$ が一致しない場合、認証結果判断部 3 7 は、エラー発生器 3 8 に対して、一致しないことを通知する。これを受けて、エラー発生器 3 8 はエラーを発生して処理を中断する。一方、認証結果判断部 3 7 において V_2 と $\text{HASH}(01 \parallel Y_1 \parallel Y_2 \parallel MK)$ が一致した場合はサーバ 2 が正当な装置として認証してセッション鍵生成器 3 9 は、 $SK = \text{HASH}(11 \parallel Y_1 \parallel Y_2 \parallel MK)$ によりセッション鍵 SK を生成する。

(4) 擬似乱数発生器を利用した場合 (その 1 とその 2)

次に、疑似乱数関数を利用した場合の端末装置 1 の動作を説明する。端末装置 1 は、前述した疑似乱数発生器を利用した場合 (その 1)、疑似乱数発生器を利用した場合 (その 2) に関わらず次のように動作する。

疑似乱数関数を利用した場合は、ユーザの端末装置 1 に備えたメモリ 1 2 に記憶されたハッシュ関数 HASH の代わりに疑似乱数関数 PRNG を用いる以外は、ハッシュ関数を利用した場合と同様の動作であるため、ここでは詳細な説明を省略する。

＜サーバの動作＞

(1) 多項式を利用した場合(その1)、多項式とハッシュ関数を利用した場合(その1)、ハッシュ関数を利用した場合(その1)、擬似乱数関数を利用した場合(その1)

サーバ2は、前述した多項式利用した場合(その1)、多項式とハッシュ関数を利用した場合(その1)、ハッシュ関数利用した場合(その1)、擬似乱数発生器を利用した場合(その1)に関わらず次のように動作する。

サーバ2に備えたメモリ41に保存されたユーザIDとパスワードの認証データHを読み出す。マスク演算器43はメモリ41から読み出したHと乱数発生器42から発生させた乱数 R_2 を入力として Y_2 を、 $Y_2 = g^{R_2} \cdot H \bmod p$ により計算する。通信処理部44は、計算して得られた Y_2 を端末装置1に送信し、端末装置1から受信した Y_1 をマスター鍵生成器45へ出力する。マスク鍵生成器45はメモリ41から読み出したHと乱数発生器42からの R_2 と通信処理部44からの Y_1 を入力としてMKを、 $MK = (Y_1 \cdot H)^{R_2} \bmod p$ により計算して、MKを出力する。

続いて認証結果判断部46は、MKを入力として、 $V_2 = \text{HASH}(01 \parallel Y_1 \parallel Y_2 \parallel MK)$ により V_2 を計算してこの V_2 を通信処理部44により端末装置1へ送信し、端末装置1から受信した V_1 と $\text{HASH}(00 \parallel Y_1 \parallel Y_2 \parallel MK)$ を比較する。ここで、HASHは一方向ハッシュ関数であるし、HASHの代わりにMAC(Message Authentication Code)を使ってもよい。

次に、認証結果判断部46において V_1 と $\text{HASH}(00 \parallel Y_1 \parallel Y_2 \parallel MK)$ が一致しない場合、認証結果判断部46は、エラー発生器47に対して、一致しないことを通知する。これを受けて、エラー発生器47はエラーを発生して処理を中断する。一方、認証結果判断部46において V_1 と $\text{HASH}(00 \parallel Y_1 \parallel Y_2 \parallel MK)$ が一致した場合は端末装置1が正当な装置として認証してセッション鍵生成器48は、 $SK = \text{HASH}(11 \parallel Y_1 \parallel Y_2 \parallel MK)$ によりセッション鍵SKを生成する。

(2) 多項式を利用した場合(その2)、多項式とハッシュ関数を利用した場合(その2)、ハッシュ関数を利用した場合(その2)、擬似乱数関数を利用した場合(その2)

サーバ2は、前述した多項式利用した場合(その2)、多項式とハッシュ関数を利用した場合(その2)、ハッシュ関数を利用した場合(その2)、擬似乱数発生器を利用した場合(その2)に関わらず次のように動作する。

サーバ2に備えたメモリ41に保存されたユーザIDとパスワードの認証データHを読み出す。マスク演算器43はメモリ41から読み出したHと乱数発生器42から発生させた乱数 R_2 を入力として Y_2 を、 $Y_2 = g^{R_2} \cdot h^H \bmod p$ により計算する。通信処理部44は、計算して得られた Y_2 を端末装置1に送信し、端末装置1から受信した Y_1 をマスター鍵生成器45へ出力する。マスク鍵生成器45はメモリ41から読み出したHと乱数発生器42からの R_2 と通信処理部44からの Y_1 を入力としてMKを、 $MK = (Y_1 \cdot h^H)^{R_2} \bmod p$ により計算して、MKを出力する。

続いて認証結果判断部46は、MKを入力として、 $V_2 = \text{HASH}(01 \parallel Y_1 \parallel Y_2 \parallel MK)$ により V_2 を計算してこの V_2 を通信処理部44により端末装置1へ送信し、端末装置1から受信した V_1 と $\text{HASH}(00 \parallel Y_1 \parallel Y_2 \parallel MK)$ を比較する。ここで、HASHは一方向ハッシュ関数であるし、HASHの代わりにMAC(Message Authentication Code)を使ってもよい。

次に、認証結果判断部46において V_1 と $\text{HASH}(00 \parallel Y_1 \parallel Y_2 \parallel MK)$ が一致しない場合、認証結果判断部46は、エラー発生器47に対して、一致しないことを通知する。これを受けて、エラー発生器47はエラーを発生して処理を中断する。一方、認証結果判断部46において V_1 と $\text{HASH}(00 \parallel Y_1 \parallel Y_2 \parallel MK)$ が一致した場合は端末装置1が正当な装置として認証してセッション鍵生成器48は、 $SK = \text{HASH}(11 \parallel Y_1 \parallel Y_2 \parallel MK)$ によりセッション鍵SKを生成する。

<パスワード認証データの更新-1>

ユーザは、サーバに対してすでに登録されたパスワード認証データを、自分が覚えているパスワードを変えずに、更新したい時、自分の端末装置の更新化を行う。図9は、ユーザの端末装置の更新化処理の構成を示すブロック図である。更新化処理は、多項式発生器13による多項式 T' とユーザの端末装置1に備えたメモリ12から記憶された多項式 P' が入力されると、更新値生成器14によって、サーバ更新用の値 H' と、ユーザ保存用の更新された多項式 P' が生成され、 H' は、サーバに受け渡され、更新された多項式 P' は、メモリ12へ保存する。ここでの更新化処理は、前述した多項式を利用した場合（その1）、多項式を利用した場合（その2）、多項式とハッシュ関数を利用した場合（その1）、多項式とハッシュ関数を利用した場合（その2）に適用することが可能である。

<端末装置の更新化処理>

（1）多項式を利用した場合（その1）

初めに、図9を参照して、多項式を利用した場合（その1）の端末装置1の更新化処理の動作を説明する。

まず、多項式発生器13によりランダムに多項式を発生する。このとき、登録したサーバの数が一つだったら x を変数とする1次多項式 $(t'(x) = \beta_1 \cdot x \bmod q)$ を、サーバの数が n 個だったら n 次多項式 $(t'(x) = \beta_1 \cdot x + \beta_2 \cdot x^2 + \dots + \beta_n \cdot x^n \bmod q)$ を発生する。ここで、 β は $(\mathbb{Z}/q\mathbb{Z})^*$ から無作為に選ばれる。例えば、一つのサーバの場合、 $T' = t'(x)$ は、 $t'(x) = \beta_1 \cdot x \bmod q$ となる。ここで、ユーザの端末装置1に備えたメモリ12から記憶された多項式 $P' = p'(x)$ を読み出す。多項式 $t'(x)$ と多項式 $p'(x)$ が入力されたら更新値生成器14は、ユーザ保存用の更新された多項式 P' とサーバ更新用の値 H' を生成する。更新された多項式 P' は、例えば $P' = t'(x) + p'(x) = (\alpha_1 + \beta_1) \cdot x \bmod q$ により計算できる。サーバ更新用の値 H' 、例えば $H' = h^{t'(1)} \bmod p$ により計算できる。ここで、 $t'(1)$ は $t'(x)$ で x の代わりにサーバのID（例えば、「1」）入れて計算した値である。サーバ更新用の値 H' はユ

一ザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ 12 は更新された多項式 $P' = t'(x) + p'(x)$ を記憶して保存する。

(2) 多項式を利用した場合 (その 2)

次に、図 9 を参照して、多項式を利用した場合 (その 1) の端末装置 1 の更新化処理の動作を説明する。

まず、多項式発生器 13 によりランダムに多項式を発生する。このとき、登録したサーバの数が一つだったら x を変数とする 1 次多項式 ($t'(x) = \beta_1 \cdot x \bmod q$) を、サーバの数が n 個だったら n 次多項式 ($t'(x) = \beta_1 \cdot x + \beta_2 \cdot x^2 + \dots + \beta_n \cdot x^n \bmod q$) を発生する。ここで、 β は $(Z/qZ)^*$ から無作為に選ばれる。例えば、一つのサーバの場合、 $T' = t'(x)$ は、 $t'(x) = \beta_1 \cdot x \bmod q$ となる。ここで、ユーザの端末装置 1 に備えたメモリ 12 から記憶された多項式 $P' = p'(x)$ を読み出す。多項式 $t'(x)$ と多項式 $p'(x)$ が入力されたら更新値生成器 14 は、ユーザ保存用の更新された多項式 P' とサーバ更新用の値 H' を生成する。更新された多項式 P' は、例えば $P' = t'(x) + p'(x) = (\alpha_1 + \beta_1) \cdot x \bmod q$ により計算できる。サーバ更新用の値 H' 、例えば $H' = t'(1) \bmod q$ により計算できる。ここで、 $t'(1)$ は $t'(x)$ で x の代わりにサーバの ID (例えば、「1」) 入れて計算した値である。サーバ更新用の値 H' はユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ 12 は更新された多項式 $P' = t'(x) + p'(x)$ を記憶して保存する。

(3) 多項式とハッシュ関数を利用した場合 (その 1)

次に、図 9 を参照して、多項式とハッシュ関数を利用した場合 (その 1) の端末装置 1 の更新化処理の動作を説明する。

まず、多項式発生器 13 によりランダムに多項式を発生する。このとき、登

録したサーバの数が一つだったら x を変数とする 1 次多項式 ($t'(x) = \beta_1 \cdot x \bmod N$) を、サーバの数が n 個だったら n 次多項式 ($t'(x) = \beta_1 \cdot x + \beta_2 \cdot x^2 + \dots + \beta_n \cdot x^n \bmod N$) を発生する。ここで、 β は $(Z/qZ)^*$ から無作為に選ばれる。例えば、一つのサーバの場合、 $T' = t'(x)$ は、 $t'(x) = \beta_1 \cdot x \bmod N$ となる。ここで、ユーザの端末装置 1 に備えたメモリ 12 から記憶された多項式とハッシュ関数 $P' = (p'(x), \text{HASH})$ を読み出す。多項式 $t'(x)$ と多項式 $p'(x)$ が入力されたら更新値生成器 14 は、ユーザ保存用の更新された多項式 P' とサーバ更新用の値 H' を生成する。更新された多項式 P' は、例えば $P' = t'(x) + p'(x) = (\alpha_1 + \beta_1) \cdot x \bmod N$ により計算できる。サーバ更新用の値 H' 、例えば $H' = h^{t'(1)} \bmod p$ により計算できる。ここで、 $t'(1)$ は $t'(x)$ で x の代わりに「1」を入れて計算した値である。

例えば、登録したサーバの数が n の場合、更新値生成器 14 は、 i 番目のサーバに対してサーバ更新用の値 H' を生成する。サーバ更新用の値 H' は、例えば $H' = h^{t'(i)} \bmod p$ により計算できる。ここで、 $t'(i)$ は n 次多項式 $t'(x)$ で x の代わりに「 i 」を入れて計算した値である。

サーバ更新用の値 H' はユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ 12 は更新された多項式 $P' = t'(x) + p'(x)$ とメモリ 12 から読み出したハッシュ関数 HASH とを一緒に $P' = (t'(x) + p'(x), \text{HASH})$ として記憶して保存する。

(4) 多項式とハッシュ関数を利用した場合 (その 2)

次に、図 9 を参照して、多項式とハッシュ関数を利用した場合 (その 2) の端末装置 1 の更新化処理の動作を説明する。

まず、多項式発生器 13 によりランダムに多項式を発生する。このとき、登録したサーバの数が一つだったら x を変数とする 1 次多項式 ($t'(x) = \beta_1 \cdot x \bmod N$) を、サーバの数が n 個だったら n 次多項式 ($t'(x) = \beta_1 \cdot x + \beta_2 \cdot x^2 + \dots + \beta_n \cdot x^n \bmod N$) を発生する。ここで、 β は $(Z/qZ)^*$

$qZ)^*$ から無作為に選ばれる。例えば、一つのサーバの場合、 $T' = t'(x)$ は、 $t'(x) = \beta_1 \cdot x \bmod N$ となる。ここで、ユーザの端末装置1に備えたメモリ12から記憶された多項式とハッシュ関数 $P' = (p'(x), \text{HASH})$ を読み出す。多項式 $t'(x)$ と多項式 $p'(x)$ が入力されたら更新値生成器14は、ユーザ保存用の更新された多項式 P' とサーバ更新用の値 H' を生成する。更新された多項式 P' は、例えば $P' = t'(x) + p'(x) = (\alpha_1 + \beta_1) \cdot x \bmod N$ により計算できる。サーバ更新用の値 H' 、例えば $H' = t'(1) \bmod N$ により計算できる。ここで、 $t'(1)$ は $t'(x)$ で x の代わりに「1」を入れて計算した値である。

例えば、登録したサーバの数が n の場合、更新値生成器14は、 i 番目のサーバに対してサーバ更新用の値 H' を生成する。サーバ更新用の値 H' は、例えば $H' = t'(i) \bmod N$ により計算できる。ここで、 $t'(i)$ は n 次多項式 $t'(x)$ で x の代わりに「 i 」を入れて計算した値である。

サーバ更新用の値 H' はユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ12は更新された多項式 $P' = t'(x) + p'(x)$ とメモリ12から読み出したハッシュ関数 HASH とを一緒に $P' = (t'(x) + p'(x), \text{HASH})$ として記憶して保存する。

<サーバの更新化処理>

(1) 多項式を利用した場合(その1)、多項式とハッシュ関数を利用した場合(その1)

初めに、図10を参照して、多項式を利用した場合(その1)、多項式とハッシュ関数を利用した場合(その1)のサーバ2の更新化処理の動作を説明する。サーバ2は、前述した多項式利用した場合(その1)、多項式とハッシュ関数を利用した場合(その1)に関わらず次のように動作する。

まず、サーバ2に備えたメモリ41に保存されたユーザIDとパスワード認証データ H を読み出す。ユーザの端末装置から送られたサーバ更新用の値 H' とメモリ41から読み出したパスワード認証データ H が入力されたら更新値

生成器 2 1 は、サーバ保存用の更新されたパスワード認証データ H を生成する。更新されたパスワード認証データ H は、例えば $H = h^{p(1)} \cdot h^{t'(1)} = h^{p(1) + t'(1)} \bmod p$ により計算できる。サーバの内部にあるメモリ 4 1 は更新されたパスワード認証データ H を記憶して保存する。

(2) 多項式を利用した場合 (その 2)

次に、図 1 0 を参照して、前述した多項式を利用した場合 (その 2) のサーバ 2 の更新化処理の動作を説明する。

まず、サーバ 2 に備えたメモリ 4 1 に保存されたユーザ ID とパスワード認証データ H を読み出す。ユーザの端末装置から送られたサーバ更新用の値 H' とメモリ 4 1 から読み出したパスワード認証データ H が入力されたら更新値生成器 2 1 は、サーバ保存用の更新されたパスワード認証データ H を生成する。更新されたパスワード認証データ H は、例えば $H = p(1) + t'(1) \bmod q$ により計算できる。サーバの内部にあるメモリ 4 1 は更新されたパスワード認証データ H を記憶して保存する。

(3) 多項式とハッシュ関数を利用した場合 (その 2)

次に、図 1 0 を参照して、前述した多項式とハッシュ関数を利用した場合 (その 2) のサーバ 2 の更新化処理の動作を説明する。

まず、サーバ 2 に備えたメモリ 4 1 に保存されたユーザ ID とパスワード認証データ H を読み出す。ユーザの端末装置から送られたサーバ更新用の値 H' とメモリ 4 1 から読み出したパスワード認証データ H が入力されたら更新値生成器 2 1 は、サーバ保存用の更新されたパスワード認証データ H を生成する。更新されたパスワード認証データ H は、例えば $H = p(1) + t'(1) \bmod N$ により計算できる。サーバの内部にあるメモリ 4 1 は更新されたパスワード認証データ H を記憶して保存する。

<パスワード認証データの更新 - 2 >

ユーザは、サーバに対してすでに登録されたパスワード認証データを、自分

が覚えているパスワードを変えながら、更新したい時、自分の端末装置の更新化処理を行う。図 1 1 は、ユーザの端末装置の更新化処理の構成を示すブロック図である。更新化処理は、秘密値発生器 1 5 による秘密値 S' とユーザの新しいパスワード PW' とユーザの端末装置 1 に備えたメモリ 1 2 から記憶された P' が入力されると、パスワード認証データ更新器 1 6 によって、サーバ更新用のパスワード認証データ H' と、ユーザ保存用の更新された P' が生成され、 H' は、サーバ 2 に受け渡され、更新された P' は、メモリ 1 2 へ保存する。ここでの更新化処理は、前述したハッシュ関数を利用した場合（その 1）、ハッシュ関数を利用した場合（その 2）、擬似乱数発生器を利用した場合（その 1）、擬似乱数発生器を利用した場合（その 2）に適用することが可能である。そして、多項式を利用した場合（その 1）、多項式を利用した場合（その 2）、多項式とハッシュ関数を利用した場合（その 1）、多項式とハッシュ関数を利用した場合（その 2）も同じように初期化処理と同様の動作を用いて適用することが可能なため、ここでは詳細な説明を省略する。

＜端末装置の更新化処理＞

（1）ハッシュ関数を利用した場合（その 1）

初めに、図 1 1 を参照して、ハッシュ関数を利用した場合（その 1）の端末装置 1 の更新化処理の動作を説明する。

まず、秘密値発生器 1 5 によりランダムに秘密値 S' を発生する。ここで、ユーザの端末装置 1 に備えたメモリ 1 2 から記憶された $P' = (S, HASH)$ を読み出す。ユーザが覚えている新しいパスワード (PW') とハッシュ関数 $HASH$ と秘密値 S' が入力されたらパスワード認証データ更新器 1 6 は、ユーザ保存用の更新された P' とサーバ更新用のパスワード認証データ H' を生成する。サーバ更新用のパスワード認証データ H' は、例えば $H' = h^{HASH(S' || PW' || ID(U) || ID(S))} \bmod p$ により計算できる。ここで、 $ID(U)$ と $ID(S)$ はそれぞれユーザとサーバの ID を表す。サーバ更新用のパスワード認証データ H' はユーザが直接にサーバ 2 に渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置 1 の内

部にあるメモリ 12 は更新された $P' = (S', \text{HASH})$ を記憶して保存する。

(2) ハッシュ関数を利用した場合 (その 2)

次に、図 11 を参照して、ハッシュ関数を利用した場合 (その 2) の端末装置 1 の更新化処理の動作を説明する。

まず、秘密値発生器 15 によりランダムに秘密値 S' を発生する。ここで、ユーザの端末装置 1 に備えたメモリ 12 から記憶された $P' = (S, \text{HASH})$ を読み出す。ユーザが覚えている新しいパスワード (PW') とハッシュ関数 HASH と秘密値 S' が入力されたらパスワード認証データ更新器 16 は、ユーザ保存用の更新された P' とサーバ更新用のパスワード認証データ H' を生成する。サーバ更新用のパスワード認証データ H' は、例えば $H' = \text{HASH}(S' \parallel PW' \parallel \text{ID}(U) \parallel \text{ID}(S)) \bmod q$ により計算できる。ここで、 $\text{ID}(U)$ と $\text{ID}(S)$ はそれぞれユーザとサーバの ID を表す。サーバ更新用のパスワード認証データ H' はユーザが直接にサーバ 2 に渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置 1 の内部にあるメモリ 12 は更新された $P' = (S', \text{HASH})$ を記憶して保存する。

(3) 擬似乱数発生器を利用した場合 (その 1)、擬似乱数発生器を利用した場合 (その 2)

次に、図 11 を参照して、擬似乱数発生器を利用した場合 (その 1)、擬似乱数発生器を利用した場合 (その 2) の端末装置 1 の更新化処理の動作を説明する。

擬似乱数関数を利用した場合 (その 1 とその 2) は、メモリ 12 に記憶されたハッシュ関数 HASH の代わりに擬似乱数関数 PRNG を用いる以外は、ハッシュ関数を利用した場合 (その 1 とその 2) と同様の動作であるため、ここでは詳細な説明を省略する。

＜サーバの更新化处理＞

(1) ハッシュ関数を利用した場合(その1)、ハッシュ関数を利用した場合(その2)、擬似乱数発生器を利用した場合(その1)、擬似乱数発生器を利用した場合(その2)

初めに、図12を参照して、ハッシュ関数を利用した場合(その1)、ハッシュ関数を利用した場合(その2)、擬似乱数発生器を利用した場合(その1)、擬似乱数発生器を利用した場合(その2)のサーバ2の更新化处理の動作を説明する。サーバ2は、前述したハッシュ関数を利用した場合(その1とその2)、擬似乱数発生器を利用した場合(その1とその2)に関わらず次のように動作する。

まず、サーバ2に備えたメモリ41に保存されたユーザIDとパスワード認証データHを読み出す。ユーザの端末装置1から送られたサーバ更新用のパスワード認証データH'とメモリ41から読み出したパスワード認証データHが入力されたらパスワード認証データ更新器22は、サーバ保存用のパスワード認証データHをユーザの端末装置1から送られたH'に更新する。サーバの内部にあるメモリ41は更新されたパスワード認証データ $H=H'$ を記憶して保存する。

＜第2実施例＞

以下、本発明の公開鍵暗号方式を使った実施例について説明する。ただし、本発明は以下の各実施例に限定されるものではなく、例えばこれら実施例の構成要素同士を適宜組み合わせてもよい。

ここで、RSA公開鍵暗号方式を使った実施例の説明する前に、いくつかの背景知識及び基礎的な記号について説明しておく。

公開鍵暗号方式では、公開鍵(Pu b K)および秘密鍵(P r i K)の対(Pu b K, P r i K)が存在する。公開鍵は秘密ではなく、誰でも得ることができる。暗号化は、公開鍵を用いてメッセージmを $C=E n c_{P u b K}(m)$ となるような暗号文Cを生成することが可能である。暗号文は、秘密鍵を用いてのみ、 $m=D e c_{P r i K}(C)$ と復号可能である。暗号文は、公開鍵を用いては

復号できない。そして公開鍵署名方式では、メッセージ m を署名化して $s = \text{Sig}_{\text{Pr i K}}(m)$ となるような署名文 (m, s) を生成することが可能である。署名文の検証は、公開鍵を用いて、 $m' = \text{Ver}_{\text{Pub K}}(s)$ を求め、 m と m' を比較することで可能である。つまり、 m と m' が一致した場合はその署名文 (m, s) が正しいことを検証する。そうではない場合は、 (m, s) が正しい署名文ではないことになる。

周知のRSA公開鍵方式では、公開鍵は (N, e) であり、秘密鍵は (N, d) である。ただし、 N は、2つのランダムに選択された大きな素数 p と q の積であり（すなわち、 $N = p \cdot q$ ）、 e は、 e と $(p-1) \cdot (q-1)$ の最大公約数が1であるような小さい任意の数（例えば、 $e = 3$ あるいは $e = 2^{16} + 1$ ）であり、 d は、 $e^{-1} \bmod ((p-1) \cdot (q-1))$ である。安全性を最大化するには、 p と q を同じ長さにする。メッセージ m ($m \in \mathbb{Z}_N^*$) に対して暗号化関数は、 $\text{Enc}_{\text{Pub K}}(m) = m^e \bmod N$ であり、復号化関数は、 $\text{Dec}_{\text{Pr i K}}(C) = C^d \bmod N$ である。ここで、暗号文 C と公開鍵 (N, e) が与えられた時、メッセージ m を求めるのは計算量的に困難である。RSAは、大きな数 N の素因数分解が難しいことから安全性を得ている。そして署名化関数は、 $\text{Sig}_{\text{Pr i K}}(m) = m^d \bmod N$ であり、検証化関数は、 $\text{Ver}_{\text{Pub K}}(s) = s^e \bmod N$ である。一般に、暗号システムでは、その安全性のレベルを記述する安全性パラメータを有する。ここでは、ハッシュ関数HASHの安全性パラメータとして k を用い（ただし、 $1/2^k$ は無視できるほど小さいと仮定する）、RSA公開鍵方式の安全性パラメータとして1を用い、特に、RSAの法 N は長さ1であると仮定する。また、 $\{0, 1\}^*$ は有限の2進数のストリングの集合を、 $\{0, 1\}^k$ は長さ k の2進数のストリングの集合を示す。ハッシュ関数HASHは $\{0, 1\}^*$ の入力から $\{0, 1\}^k$ の出力を出す安全な一方向関数であり、FDH (Full-Domain Hash) 関数は $\{0, 1\}^*$ の入力から $\mathbb{Z}_N^* \setminus \{1\}$ の出力を出す安全な一方向関数である。また、乱数発生器から発生される乱数は T ($T \in \mathbb{Z}_N^*$) を無作為に生成する。また、 \parallel は値を連結 (concatenation) するという意味である。

＜端末装置の初期化＞

ユーザは、サーバに対して個人登録したい時、自分の端末装置の初期化を行う。図1は、ユーザの端末装置の初期化処理の構成を示すブロック図である。初期化は、ユーザがパスワードを入力すると、データ伸張器11によって、サーバ登録用のパスワード認証データHと、ユーザ保存用の値P'が生成され、パスワード認証データHは、サーバに受け渡され、値P'は、メモリ12へ保存する。ここで、データ伸張器11は、多項式とFDH関数、FDH関数などで構成することが可能である。

(1) 多項式とFDH関数を利用する場合 (その1)

初めに、図13を参照して、多項式とFDH関数を利用する場合 (その1) について説明する。

まず、FDH関数発生器122によりランダムにFDH関数FDHを発生する。そして、多項式発生器123によりランダムに多項式を発生する。このとき、登録するサーバの数が一つだったら x を変数とする1次多項式($p'(x) = \alpha_1 \cdot x \bmod N$)を、サーバの数が n 個だったら n 次多項式($p'(x) = \alpha_1 \cdot x + \alpha_2 \cdot x^2 + \dots + \alpha_n \cdot x^n \bmod N$)を発生する。ここで、 α は Z_N^* から無作為に選ばれる。例えば、一つのサーバの場合、 $p'(x)$ は、 $p'(x) = \alpha_1 \cdot x \bmod N$ となる。ここで、ユーザは自分が覚えているパスワード (例えば、"P o o h 9 3") を入力する。多項式とFDH関数とユーザのパスワードが入力されたらパスワード認証データ生成器124は、パスワード認証データHを生成する。パスワード認証データHは、例えば $H = p(1) = p'(1) + P o o h 9 3 \bmod N$ により計算できる。ここで、 $p'(1)$ は $p'(x)$ で x の代わりにサーバのID (例えば、「1」) を入れて計算した値である。パスワード認証データHはユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ12は多項式発生器から発生された多項式 $p'(x)$ とFDH関数発生器から発生されたFDH関数FDH

とを一緒に $P' = (p'(x), FDH)$ として記憶して保存する。

(2) 多項式とFDH関数を利用する場合 (その2)

次に、図13を参照して、多項式とFDH関数を利用する場合 (その2) を利用する場合について説明する。

まず、FDH関数発生器122によりランダムにFDH関数FDHを発生する。そして、多項式発生器123によりランダムに多項式を発生する。このとき、登録するサーバの数が一つだったら x を変数とする1次多項式 $(p'(x) = \alpha_1 \cdot x \bmod N)$ を、サーバの数が n 個だったら n 次多項式 $(p'(x) = \alpha_1 \cdot x + \alpha_2 \cdot x^2 + \dots + \alpha_n \cdot x^n \bmod N)$ を発生する。ここで、 α は Z_N^* から無作為に選ばれる。例えば、一つのサーバの場合、 $p'(x)$ は、 $p'(x) = \alpha_1 \cdot x \bmod N$ となる。ここで、ユーザは自分が覚えているパスワード (例えば、"P o o h 9 3") を入力する。多項式とFDH関数とユーザのパスワードが入力されたらパスワード認証データ生成器124は、パスワード認証データ H を生成する。パスワード認証データ H は、例えば $H = p(1) = p'(1) + FDH(P o o h 9 3 \parallel ID(U) \parallel ID(S)) \bmod N$ により計算できる。ここで、 $ID(U)$ と $ID(S)$ はそれぞれユーザとサーバの ID を表す。ここで、 $p'(1)$ は $p'(x)$ で x の代わりに「1」を入れて計算した値である。

例えば、登録するサーバの数が n の場合、パスワード認証データ生成器124は、 i 番目のサーバに対してパスワード認証データ H を生成する。パスワード認証データ H は、例えば $H = p(i) = p'(i) + FDH(P o o h 9 3 \parallel ID(U) \parallel ID(S)) \bmod N$ により計算できる。ここで、 $ID(U)$ と $ID(S)$ はそれぞれユーザと i 番目のサーバの ID を表す。ここで、 $p'(i)$ は n 次多項式 $p'(x)$ で x の代わりに「 i 」を入れて計算した値である。

パスワード認証データ H はユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ12は多項式発生器から発生された多項式 p'

(x) とFDH関数発生器から発生されたFDH関数FDHとを一緒に $P' = (p'(x), FDH)$ として記憶して保存する。

(3) FDH関数を利用する場合

次に、図14を参照して、FDH関数を利用する場合について説明する。

まず、FDH関数発生器125によりランダムにFDH関数FDHを発生する。そして、秘密値発生器126によりランダムに秘密値Sを発生する。ただし、Sは全探索攻撃を防げるような長さの値である（例えば、Sは80ビット以上の値）。ここで、ユーザは自分が覚えているパスワード（例えば、"P o o h 9 3"）を入力する。FDH関数と秘密値Sとユーザのパスワードが入力されたらパスワード認証データ生成器127はパスワード認証データHを生成する。パスワード認証データHは、例えば、 $H = FDH(S \parallel P o o h 9 3 \parallel ID(U) \parallel ID(S))$ により計算できる。ここで、ID(U)とID(S)はそれぞれユーザとサーバのIDを表す。パスワード認証データHはユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ12は秘密値発生器126から発生された秘密値SとFDH関数発生器125から発生されたFDH関数FDHとを一緒に $P' = (S, FDH)$ として記憶して保存する。

<端末装置とサーバとの初期化>

サーバは、RSA公開鍵をユーザに送信したい時、初期化処理を行う。サーバは、RSA公開鍵方式に従って公開鍵と秘密鍵の対を生成し、その公開鍵をユーザに送信する。ここで、初期化は、安全な通信、安全ではない通信などで実現することが可能である。安全ではない通信を利用する場合では、ユーザは、受信した公開鍵が正しいかどうかを決定する。このユーザによる決定は、ユーザに、サーバが適当な方法で選択した公開鍵を生成したかどうかを決定する方法を提供する。つまり、サーバが提供した公開鍵eと $(p-1) \cdot (q-1)$ の最大公約数が1で（すなわち、 $gcd(e, (p-1) \cdot (q-1))$ ）

= 1) あるかどうかを決定するためにRSA署名方式を使う。

(1) 安全な通信を利用する場合

初めに、図15を参照して、安全な通信を利用する場合の初期化処理について説明する。

まず、RSA鍵生成器23により公開鍵(N, e)と秘密鍵(N, d)の対を生成する。RSA公開鍵(N, e)はサーバが直接にユーザに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。サーバの内部にあるメモリ41はRSA秘密鍵(N, d)を記憶して保存する。

(2) 安全ではない通信を利用する場合

次に、図16を参照して、安全ではない通信を利用する場合の初期化処理について説明する。

まず、ユーザの端末装置1にある乱数発生器17によりランダムに乱数 R_1 ($R_1 \in \{0, 1\}^k$)を発生し、サーバに送信する。一方、サーバ2にあるRSA鍵生成器24により公開鍵(N, e)と秘密鍵(N, d)の対を生成する。そして、乱数発生器25によりランダムに乱数 R_2 ($R_2 \in \{0, 1\}^k$)を発生する。RSA署名文生成器26は、端末装置1から受信した R_1 とRSA鍵生成器24から生成された秘密鍵(N, d)と乱数発生器25から発生させた乱数 R_2 を入力として、 $\{m_j\}_{1 \leq j \leq n}$ の署名 $\{s_j\}_{1 \leq j \leq n}$ を生成する(ただし、 n は $n \geq \log_e(PW \cdot (e-1)/e)$ になるような整数である。ここで、 PW はパスワードを表す。)。署名 $\{s_j\}_{1 \leq j \leq n}$ は、 $\{s_j = m_j^d \bmod N\}_{1 \leq j \leq n}$ により計算する。ここで、 $\{m_j\}_{1 \leq j \leq n}$ は $\text{HASH}(n \parallel N \parallel e \parallel \text{ID}(U) \parallel \text{ID}(S) \parallel R_1 \parallel R_2)$ の出力から長さ1のブロックを n 個に分けて得たものである。ID(U)とID(S)はそれぞれユーザとサーバのIDを表す。サーバ2は、公開鍵(N, e)と計算して得られた署名文($R_2, \{s_j\}_{1 \leq j \leq n}$)を端末装置1に送信する。サーバ2の内部に備えたメモリ41は、RSA鍵生成器24から生成された秘密鍵(N, d)を記憶して保存する。

ユーザの端末装置 1 にある検証結果判断部 18 は、乱数発生器 17 において発生させた R_1 とサーバ 2 から受信した $((N, e), (R_2, \{s_j\}_{1 \leq j \leq n}))$ を入力として、署名文 $(R_2, \{s_j\}_{1 \leq j \leq n})$ の検証を行う。 $\{m_j = s_j^e \bmod N\}_{1 \leq j \leq n}$ により $\{m_j\}_{1 \leq j \leq n}$ を計算してこの $\{m_j\}_{1 \leq j \leq n}$ と $\text{HASH}(n \parallel N \parallel e \parallel \text{ID}(U) \parallel \text{ID}(S) \parallel R_1 \parallel R_2)$ を比較する。検証結果判断部 18 において $\{m_j\}_{1 \leq j \leq n}$ と $\text{HASH}(n \parallel N \parallel e \parallel \text{ID}(U) \parallel \text{ID}(S) \parallel R_1 \parallel R_2)$ が一致しない場合、検証結果判断部 18 は、エラー発生器 19 に対して、一致しないことを通知する。これを受けて、エラー発生器 19 はエラーを発生して処理を中断する。一方、検証結果判断部 18 において $\{m_j\}_{1 \leq j \leq n}$ と $\text{HASH}(n \parallel N \parallel e \parallel \text{ID}(U) \parallel \text{ID}(S) \parallel R_1 \parallel R_2)$ が一致した場合はサーバ 2 の公開鍵 (N, e) が正当なものとして検証して、ユーザの端末装置 1 の内部に備えたメモリ 12 に、公開鍵 (N, e) を記憶して保存する。

次に、図 17、18 を参照して、前述した初期化を行った端末装置 1 とサーバ 2 (図 5 参照) との間で相互認証及び鍵交換を行う動作を説明する。

<端末装置の動作>

(1) 多項式と FDH 関数を利用した場合 (その 1)

初めに、多項式と FDH 関数を利用した場合 (その 1) の端末装置 1 の動作を説明する。

まず、ユーザの端末装置 1 に備えたメモリ 12 から記憶された多項式と FDH 関数 $P' = (p'(x), \text{FDH})$ を読み出す。結合器 52 はメモリ 12 から読み出した多項式 $p'(x)$ と FDH 関数 FDH とユーザが入力したパスワードにより $W = \text{FDH}(p(x) \parallel \text{ID}(U) \parallel \text{ID}(S))$ を計算して出力する。ここで、 $p(x) = p'(x) + \text{Pooh93} \bmod N$ により計算する。例えば、 $p'(x)$ が 1 次多項式の場合、 $p(x) = p(1) = p'(1) + \text{Pooh93} = \alpha_1 \cdot 1 + \text{Pooh93} \bmod N$ により計算する。ここで、 $p'(1)$ は $p'(x)$ で x の代わりに「1」入れて計算した値である。

ユーザの端末装置 1 に備えたメモリ 12 から読み出した多項式 $p'(x)$ が n 次多項式の場合、結合器 52 は多項式 $p'(x)$ と FDH 関数 FDH とユー

ザが入力したパスワードにより $W = \text{FDH}(p(x) \parallel \text{ID}(U) \parallel \text{ID}(S))$ を計算して出力する。ここで、 $p(x) = p'(x) + \text{Pooh93} \bmod N$ により計算する。例えば、 $p(x) = p(i) = p'(i) + \text{Pooh93} \bmod N$ により計算する。ここで、 $p'(i)$ は i 番目のサーバに対して $p'(x)$ で x の代わりに「 i 」入れて計算した値である。

マスク演算器 54 は、メモリ 12 から読み出した公開鍵 (N, e) と結合器 52 から入力された W と乱数発生器 53 においてランダムに発生させた乱数 T ($T \in \mathbb{Z}_N^*$) とから Z を、 $Z = T^e \cdot W \bmod N$ により計算する。通信処理部 55 は Z をサーバ 2 へ送信し、サーバ 2 から V_2 を受信する。

続いて認証結果判断部 56 は、乱数発生器 53 から出力された T を入力として、 $\text{HASH}(01 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ を計算してサーバ 2 から受信した V_2 と比較する。ここで、 HASH の代わりに MAC を使ってもよい。認証結果判断部 56 において V_2 と $\text{HASH}(01 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ が一致しない場合、認証結果判断部 56 は、エラー発生器 57 に対して、一致しないことを通知する。これを受けて、エラー発生器 57 はエラーを発生して処理を中断する。一方、認証結果判断部 56 において V_2 と $\text{HASH}(01 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ が一致した場合はサーバ 2 が正当な装置として認証して、検証子生成器 58 は、 $V_1 = \text{HASH}(00 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ により検証子 V_1 を計算してサーバ 2 へ送信する。同時に、セッション鍵生成器 59 は、 $SK = \text{HASH}(11 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ によりセッション鍵 SK を生成する。

(2) 多項式と FDH 関数を利用した場合 (その 2)

次に、多項式と FDH 関数を利用した場合 (その 2) の端末装置 1 の動作を説明する。

まず、ユーザの端末装置 1 に備えたメモリ 12 から記憶された多項式と FDH 関数 $P' = (p'(x), \text{FDH})$ を読み出す。結合器 52 はメモリ 12 から読み出した多項式 $p'(x)$ と FDH 関数 FDH とユーザが入力したパスワードにより $W = \text{FDH}(p(x) \parallel \text{ID}(U) \parallel \text{ID}(S))$ を計算して出力

する。ここで、 $p(x) = p'(x) + \text{FDH}(\text{P o o h 9 3} \parallel \text{ID}(U) \parallel \text{ID}(S)) \bmod N$ により計算する。例えば、 $p'(x)$ が1次多項式の場合、 $p(x) = p(1) = p'(1) + \text{FDH}(\text{P o o h 9 3} \parallel \text{ID}(U) \parallel \text{ID}(S)) \bmod N$ により計算する。ここで、 $p'(1)$ は $p'(x)$ で x の代わりに「1」入れて計算した値である。

ユーザの端末装置1に備えたメモリ12から読み出した多項式 $p'(x)$ が n 次多項式の場合、結合器52は多項式 $p'(x)$ とFDH関数FDHとユーザが入力したパスワードにより $W = \text{FDH}(p(x) \parallel \text{ID}(U) \parallel \text{ID}(S))$ を計算して出力する。ここで、 $p(x) = p'(x) + \text{FDH}(\text{P o o h 9 3} \parallel \text{ID}(U) \parallel \text{ID}(S)) \bmod N$ により計算する。例えば、 $p(x) = p(i) = p'(i) + \text{FDH}(\text{P o o h 9 3} \parallel \text{ID}(U) \parallel \text{ID}(S)) \bmod N$ により計算する。ここで、 $p'(i)$ は i 番目のサーバに対して $p'(x)$ で x の代わりに「 i 」入れて計算した値である。

マスク演算器54は、メモリ12から読み出した公開鍵 (N, e) と結合器52から入力された W と乱数発生器53においてランダムに発生させた乱数 T ($T \in \mathbb{Z}_N^*$) とから Z を、 $Z = T^e \cdot W \bmod N$ により計算する。通信処理部55は Z をサーバ2へ送信し、サーバ2から V_2 を受信する。

続いて認証結果判断部56は、乱数発生器53から出力された T を入力として、 $\text{HASH}(01 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ を計算してサーバ2から受信した V_2 と比較する。ここで、HASHの代わりにMACを使ってもよい。認証結果判断部56において V_2 と $\text{HASH}(01 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ が一致しない場合、認証結果判断部56は、エラー発生器57に対して、一致しないことを通知する。これを受けて、エラー発生器57はエラーを発生して処理を中断する。一方、認証結果判断部56において V_2 と $\text{HASH}(01 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ が一致した場合はサーバ2が正当な装置として認証して、検証子生成器58は、 $V_1 = \text{HASH}(00 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ により検証子 V_1 を計算してサーバ2へ送信する。同時に、セッション鍵生成器59は、 $SK = \text{HASH}(11 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$

によりセッション鍵 SK を生成する。

(3) FDH関数を利用した場合

次に、FDH関数を利用した場合の端末装置1の動作を説明する。

まず、ユーザの端末装置1に備えたメモリ12から記憶された秘密値とFDH関数 $P' = (S, FDH)$ を読み出す。結合器52はメモリ12から読み出した秘密値 S とFDH関数 FDH とユーザが入力したパスワードにより W を計算して出力する。例えば、 W は、 $W = FDH(S \parallel P \circ o h 9 3 \parallel ID(U) \parallel ID(S))$ により計算する。マスク演算器54は、メモリ12から読み出した公開鍵 (N, e) と結合器52から入力された W と乱数発生器53においてランダムに発生させた乱数 T ($T \in Z_N^*$) とから Z を、 $Z = T^e \cdot W \bmod N$ により計算する。通信処理部55は Z をサーバ2へ送信し、サーバ2から V_2 を受信する。

続いて認証結果判断部56は、乱数発生器53から出力された T を入力として、 $HASH(01 \parallel T \parallel ID(U) \parallel ID(S))$ を計算してサーバ2から受信した V_2 と比較する。ここで、 $HASH$ の代わりに MAC を使ってもよい。認証結果判断部56において V_2 と $HASH(01 \parallel T \parallel ID(U) \parallel ID(S))$ が一致しない場合、認証結果判断部56は、エラー発生器57に対して、一致しないことを通知する。これを受けて、エラー発生器57はエラーを発生して処理を中断する。一方、認証結果判断部56において V_2 と $HASH(01 \parallel T \parallel ID(U) \parallel ID(S))$ が一致した場合はサーバ2が正当な装置として認証して、検証子生成器58は、 $V_1 = HASH(00 \parallel T \parallel ID(U) \parallel ID(S))$ により検証子 V_1 を計算してサーバ2へ送信する。同時に、セッション鍵生成器59は、 $SK = HASH(11 \parallel T \parallel ID(U) \parallel ID(S))$ によりセッション鍵 SK を生成する。

<サーバの動作>

(1) 多項式とFDH関数を利用した場合(その1)、多項式とFDH関数を利用した場合(その2)

サーバ2は、前述した多項式とFDH関数を利用した場合（その1）、多項式とFDH関数を利用した場合（その2）に関わらず次のように動作する。

まず、サーバ2に備えたメモリ41から保存されたユーザIDとパスワードの認証データHを読み出す。マスター鍵生成器62はメモリ41から読み出したHと秘密鍵(N, d)と端末装置1から受信したZを入力としてTを、 $T = (Z/W)^d \bmod N$ により計算して出力する。ここで、Wは、 $W = \text{FDH}(H \parallel \text{ID}(U) \parallel \text{ID}(S))$ により計算する。検証子生成器63は、マスター鍵生成器62から入力されたTから検証子 V_2 を、 $V_2 = \text{HASH}(01 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ により計算する。通信処理部64は、計算して得られた V_2 を端末装置1へ送信し、端末装置1から受信した V_1 を認証結果判断部65へ出力する。

続いて認証結果判断部65は、マスター鍵生成器62から出力されたTを入力として、 $\text{HASH}(00 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ を計算して端末装置1から受信した V_1 と比較する。ここで、HASHの代わりにMACを使ってもよい。認証結果判断部65において V_1 と $\text{HASH}(00 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ が一致しない場合、認証結果判断部65は、エラー発生器66に対して、一致しないことを通知する。これを受けて、エラー発生器66はエラーを発生して処理を中断する。一方、認証結果判断部65において V_1 と $\text{HASH}(00 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ が一致した場合は端末装置1が正当な装置として認証して、セッション鍵生成器67は、 $SK = \text{HASH}(11 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ によりセッション鍵SKを生成する。

(2) FDH関数を利用した場合

次に、FDH関数を利用した場合のサーバ2の動作を説明する。

まず、サーバ2に備えたメモリ41から保存されたユーザIDとパスワードの認証データHを読み出す。マスター鍵生成器62はメモリ41から読み出したHと秘密鍵(N, d)と端末装置1から受信したZを入力としてTを、 $T = (Z/W)^d \bmod N$ により計算して出力する。ここで、Wは、 $W = H$ である。検証子生成器63は、マスター鍵生成器62から入力されたTから検証子 V_2

を、 $V_2 = \text{HASH}(01 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ により計算する。通信処理部64は、計算して得られた V_2 を端末装置1へ送信し、端末装置1から受信した V_1 を認証結果判断部65へ出力する。

続いて認証結果判断部65は、マスター鍵生成器62から出力された T を入力として、 $\text{HASH}(00 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ を計算して端末装置1から受信した V_1 と比較する。ここで、 HASH の代わりに MAC を使ってもよい。認証結果判断部65において V_1 と $\text{HASH}(00 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ が一致しない場合、認証結果判断部65は、エラー発生器66に対して、一致しないことを通知する。これを受けて、エラー発生器66はエラーを発生して処理を中断する。一方、認証結果判断部65において V_1 と $\text{HASH}(00 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ が一致した場合は端末装置1が正当な装置として認証して、セッション鍵生成器67は、 $SK = \text{HASH}(11 \parallel T \parallel \text{ID}(U) \parallel \text{ID}(S))$ によりセッション鍵 SK を生成する。

<パスワード認証データの更新－1>

ユーザは、サーバに対してすでに登録されたパスワード認証データを、自分が覚えているパスワードを変えずに、更新したい時、自分の端末装置の更新化を行う。図9、19は、ユーザの端末装置の更新化処理の構成を示すブロック図である。ここでの更新化処理は、前述した多項式と FDH 関数を利用した場合（その1）、多項式と FDH 関数を利用した場合（その2）とマスター鍵を利用する場合に適用することが可能である。また、この更新化処理によりサーバに対する replay 攻撃を防ぐことが可能である。

<端末装置の更新化処理>

（1）多項式と FDH 関数を利用した場合（その1）、多項式と FDH 関数を利用した場合（その2）

初めに、図9を参照して、多項式と FDH 関数を利用した場合（その1）、多項式と FDH 関数を利用した場合（その2）の端末装置1の更新化処理を説明する。端末装置1は、前述した多項式と FDH 関数を利用した場合（その1）、

多項式とFDH関数を利用した場合(その2)に関わらず次のように動作する。

まず、多項式発生器13によりランダムに多項式を発生する。このとき、登録したサーバの数が一つだったら x を変数とする1次多項式($t'(x) = \beta_1 \cdot x \bmod N$)を、サーバの数が n 個だったら n 次多項式($t'(x) = \beta_1 \cdot x + \beta_2 \cdot x^2 + \dots + \beta_n \cdot x^n \bmod N$)を発生する。ここで、 β は Z_N^* から無作為に選ばれる。例えば、一つのサーバの場合、 $T' = t'(x)$ は、 $t'(x) = \beta_1 \cdot x \bmod N$ となる。ここで、ユーザの端末装置1に備えたメモリ12から記憶された多項式とFDH関数 $P' = (p'(x), \text{FDH})$ を読み出す。多項式 $t'(x)$ と多項式 $p'(x)$ が入力されたら更新値生成器14は、ユーザ保存用の更新された多項式 P' とサーバ更新用の値 H' を生成する。更新された多項式 P' は、例えば $P' = t'(x) + p'(x) = (\alpha_1 + \beta_1) \cdot x \bmod N$ により計算できる。サーバ更新用の値 H' 、例えば $H' = t'(1) \bmod N$ により計算できる。ここで、 $t'(1)$ は $t'(x)$ で x の代わりに「1」を入れて計算した値である。

例えば、登録したサーバの数が n の場合、更新値生成器14は、 i 番目のサーバに対してサーバ更新用の値 H' を生成する。サーバ更新用の値 H' は、例えば $H' = t'(i) \bmod N$ により計算できる。ここで、 $t'(i)$ は n 次多項式 $t'(x)$ で x の代わりに「 i 」を入れて計算した値である。

サーバ更新用の値 H' はユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ12は更新された多項式 $P' = t'(x) + p'(x)$ とメモリ12から読み出したFDH関数FDHとを一緒に $P' = (t'(x) + p'(x), \text{FDH})$ として記憶して保存する。

(2) マスター鍵を利用する場合

次に、図19を参照して、マスター鍵を利用する場合の端末装置1の更新化処理の動作を説明する。

まず、乱数発生器53によりランダムに乱数 T ($T \in Z_N^*$)を発生する。ここで、ユーザの端末装置1に備えたメモリ12から記憶された多項式とFDH

関数 $P' = (p'(x), FDH)$ を読み出す。乱数 T と多項式 $p'(x)$ が入力されたら更新値生成器 20 は、ユーザ保存用の更新された多項式 P' を生成する。更新された多項式 P' は、 $P' = T + p'(x) \bmod N$ により計算できる。ユーザの端末装置の内部にあるメモリ 12 は更新された多項式 $P' = T + p'(x)$ とメモリ 12 から読み出した FDH 関数 FDH とを一緒に $P' = (T + p'(x), FDH)$ として記憶して保存する。

<サーバの更新化処理>

(1) 多項式と FDH 関数を利用した場合 (その 1)、多項式と FDH 関数を利用した場合 (その 2)

初めに、図 10 を参照して、多項式と FDH 関数を利用した場合 (その 1)、多項式と FDH 関数を利用した場合 (その 2) のサーバ 2 の更新化処理の動作を説明する。サーバ 2 は、前述した多項式と FDH 関数を利用した場合 (その 1)、多項式と FDH 関数を利用した場合 (その 2) に関わらず次のように動作する。

まず、サーバ 2 に備えたメモリ 41 に保存されたユーザ ID とパスワード認証データ H を読み出す。ユーザの端末装置から送られたサーバ更新用の値 H' とメモリ 41 から読み出したパスワード認証データ H が入力されたら更新値生成器 21 は、サーバ保存用の更新されたパスワード認証データ H を生成する。更新されたパスワード認証データ H は、例えば $H = p(1) + t'(1) \bmod N$ により計算できる。サーバの内部にあるメモリ 41 は更新されたパスワード認証データ H を記憶して保存する。

(2) マスター鍵を利用する場合

次に、図 20 を参照して、マスター鍵を利用する場合のサーバ 2 の更新化処理の動作を説明する。

まず、マスター鍵生成器 62 によりマスター鍵 T を生成する。ここで、サーバ 2 に備えたメモリ 41 から保存されたユーザ ID とパスワード認証データ H を読み出す。マスター鍵 T とメモリ 41 から読み出したパスワード認証デー

タHが入力されたら更新値生成器27は、サーバ保存用の更新されたパスワード認証データHを生成する。更新されたパスワード認証データHは、 $H = p(1) + T \bmod N$ により計算できる。サーバの内部にあるメモリ41は更新されたパスワード認証データHを記憶して保存する。

＜パスワード認証データの更新－2＞

ユーザは、サーバに対してすでに登録されたパスワード認証データを、自分が覚えているパスワードを変えながら、更新したい時、自分の端末装置の更新化処理を行う。図11は、ユーザの端末装置の更新化処理の構成を示すブロック図である。更新化処理は、秘密値発生器15による秘密値S'とユーザの新しいパスワードPW'とユーザの端末装置1に備えたメモリ12から記憶されたP'が入力されると、パスワード認証データ更新器16によって、サーバ更新用のパスワード認証データH'と、ユーザ保存用の更新されたP'が生成され、H'は、サーバ2に受け渡され、更新されたP'は、メモリ12へ保存する。ここでの更新化処理は、前述したFDH関数を利用した場合に適用することが可能である。また、この更新化処理によりサーバに対する replay 攻撃を防ぐことが可能である。そして、多項式とFDH関数を利用した場合（その1）、多項式とFDH関数を利用した場合（その2）も同じように初期化処理と同様の動作を用いて適用することが可能なため、ここでは詳細な説明を省略する。

＜端末装置の更新化処理＞

（1）FDH関数を利用した場合

図11を参照して、FDH関数を利用した場合の端末装置1の更新化処理の動作を説明する。

まず、秘密値発生器15によりランダムに秘密値S'を発生する。ここで、ユーザの端末装置1に備えたメモリ12から記憶された $P' = (S, FDH)$ を読み出す。ユーザが覚えている新しいパスワード(PW')とFDH関数FDHと秘密値S'が入力されたらパスワード認証データ更新器16は、ユーザ保存用の更新されたP'とサーバ更新用のパスワード認証データH'を生成する。サーバ

更新用のパスワード認証データ H' は、例えば $H' = FDH(S' \parallel PW' \parallel ID(U) \parallel ID(S))$ により計算できる。ここで、 $ID(U)$ と $ID(S)$ はそれぞれユーザとサーバのIDを表す。サーバ更新用のパスワード認証データ H' はユーザが直接にサーバ2に渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置1の内部にあるメモリ12は更新された $P' = (S', FDH)$ を記憶して保存する。

<サーバの更新化処理>

(1) FDH関数を利用した場合

図12を参照して、FDH関数を利用した場合のサーバ2の更新化処理の動作を説明する。

まず、サーバ2に備えたメモリ41に保存されたユーザIDとパスワード認証データ H を読み出す。ユーザの端末装置1から送られたサーバ更新用のパスワード認証データ H' とメモリ41から読み出したパスワード認証データ H が入力されたらパスワード認証データ更新器22は、サーバ保存用のパスワード認証データ H をユーザの端末装置1から送られた H' に更新する。サーバの内部にあるメモリ41は更新されたパスワード認証データ $H = H'$ を記憶して保存する。

このように、多項式を利用することにより、不正利用しようと思っている者が他人の端末装置を持っていたとしてもユーザのパスワードは情報理論的に安全である。また、サーバ内に侵入して保存されている情報を得たとしてもユーザのパスワードは情報理論的に安全である。また、ハッシュ関数と擬似乱数発生器とFDH関数を利用する場合には、不正利用しようと思っている者にとってユーザのパスワードは計算量的に安全である。

次に、前述した技術を応用した装置について説明する。

<端末に分散データを保存しない場合の遠隔分散保存装置におけるデータ保存処理>

次に、図 2 1 を参照して、端末に分散データを保存しない場合の遠隔分散保存装置におけるデータ保存処理について説明する。図 2 1 は、端末に分散データを保存しない場合の遠隔分散保存装置 5 の構成を示すブロック図である。

ユーザは分散保存したいデータ DATA を自分の端末装置 2 1 で処理し、 n 台のサーバに保存するデータ $S' 1, \dots, S' n$ へと分割する。分割データ $S' i$ はデータの ID である DID と共にユーザの端末装置 2 1 によりサーバの認証装置と共有された鍵 $SK i$ を用いて作成された安全な通信路を使用可能な通信器 5 2 によってサーバ ID i へと送られ保存される。同様の方法により、保存データの一覧情報を分割し、サーバに保存することも可能である。また、ユーザの端末装置（認証データ更新モード）を適当な間隔（記憶情報がオフライン全数探索で求まる間隔より短い間隔で、例えば、認証を行う度や 2, 3 日に 1 回など）で動作させ、 P' と各サーバに記録されている H を更新する情報 $UP', UH 1, \dots, UH n$ を生成させ、それらを更新する。

これにより、サーバに保存されるデータと認証用データをともに、漏洩と破損に強くなるように構成することができる。漏洩と破損への耐性は $(n, DS, LS 1, LS 2)$ の 4 組のパラメータで表現できる。 $DS, LS 1, LS 2$ はともに（漏洩や破損の起こる）エンティティの組み合わせの集合であり、 DS はデータの破損に対する耐性、 $LS 1, LS 2$ は漏洩に対する耐性を表現している。 DS には、壊れても構わないエンティティの組み合わせが記述される。災害などの理由によりローカルバックアップを含め保存しているデータが完全に利用できなくなったとしても、利用者が自分のデータを復元できる範囲の破損組み合わせが記述される。 $LS 1$ には、記録情報が漏れても構わないエンティティの組み合わせが記述される。保存データが漏洩したとしても、攻撃者が利用者のデータを復元することが困難な範囲の漏洩組み合わせとする。 $LS 2$ には、記録情報が漏れたとしても、後で何らかの対抗策がとれる範囲のエンティティの組み合わせが記述される。保存データが漏洩したとしても、攻撃者が利用者のデータを復元することを困難にする対抗策の存在する範囲の漏洩組み合わせとする。

また、ユーザの記憶情報はオフラインで全数探索できる程少量であると仮定

すると、従来の漏洩に弱い認証方式を使う場合、攻撃者は漏洩情報と通信路上で得た情報を使って利用者の記憶情報を全数探索でき、結果として、利用者に成りすまして遠隔分散保存されているデータをすべて入手できる。つまり、LS1にサーバ{S}とユーザの所有物{U}をそれぞれ含めることはできなかった。これに対して、漏洩に強い認証方式を使うことでLS1にサーバ{S}とユーザの所有物{U}をそれぞれ含めることが可能となる。漏洩に強い認証方式を含む全ての遠隔認証方式はユーザの所有物{U}とサーバ{S}の両方から情報が漏洩すると利用者の記憶情報を全数探索でき、結果として、利用者に成りすまして遠隔分散保存されているデータをすべて入手できる。よって、LS2にユーザの所有物{U}とサーバ{S}の組み合わせ{US}を入れることはできなかった。これに対して、 P' と H_1, \dots, H_n を更新することでLS2にユーザの所有物とサーバの組み合わせを入れることが可能となる。

次に、図22を参照して、図21に示すデータ分散器51の構成を説明する。調整器511は入力されたパラメータ n, k を秘密分散器512に渡す。秘密分散器512は入力されたパラメータ n, k に従い保存データDATAを (k, n) 分散データ S_1, S_2, \dots, S_n に変換する。次に、調整器511はデータのIDであるDIDからデータ伸長器513の入力 x を生成しデータ伸長器513へ渡す。データ伸長器513は対応する情報 H を出力し、それを暗号器514に渡す。ここでの H はオフラインでの全数探索に強い長さのものを利用する。データ伸長器513が短い H を出力する場合には異なる x を複数データ伸長器513に渡し、得られた複数の H を利用する。暗号器514は、 H を鍵として $n - k + 1$ 個以上の分散データを暗号化する。また、 S_1, \dots, S_{n-k+1} のそれぞれに改ざん検出符号を付けてもよい。暗号器の出力が S'_1, \dots, S'_n であり、DID、 ID_1, ID_2, \dots, ID_n と共にデータ分割器51の出力となる。

なお、 (k, n) 分散データとは、元データが n 個に分割されたデータであり、かつ、そのどの k 個からでも元のデータを復元できるが k 未満からでは復元できないような性質を有するデータである。 (k, n) 分散データ以外にも

任意のアクセス構造を持った分散データを利用することが可能である。また、秘密分散器は多項式や行列を使った情報量的に安全な分散方式以外に、暗号化を使うことで保存データサイズを少なくできる計算量的に安全な分散方式を用いることが可能である。

このデータ分割器 51 は、 $(n, DS, LS1, LS2) = (n, \{CS_{n-k}\}, \{UC, CS_n\}, \{UCS_{k-1}\})$ を実現している。 $\{CS\}$ はクライアントおよびサーバの記録情報およびその部分情報を意味し、 $\{S_n\}$ は n 台のサーバの全記録情報およびその部分情報を意味する。 $\{C, S\}$ は「クライアントに記録されている全情報およびその部分情報あるいはサーバに記録されている全情報およびその部分情報」を意味する。 $\{UCS_{k-1}\}$ からの漏洩は認証情報のアップデート処理で対処できる。攻撃者が $\{UCS_{k-1}\}$ から記憶情報を求める間に認証情報がアップデートされれば攻撃者は利用者のデータを求めることができなくなる。ユーザの所有物 $\{U\}$ の紛失 (P' の紛失) に対しては、 $\{U\}$ に記録してあるデータをアップデートするたびにそのコピーをローカルに作成することで解決できる。災害などでローカルコピーもろとも $\{U\}$ が破損する危険性に対しては、 $\{U\}$ に記録してあるデータを秘密分散器で (k', n) 分散データで分割して各サーバにも保存しておくことで解決できる。 $k' \geq k$ の場合、 $(n, DS, LS1, LS2) = (n, \{UCS_{n-k'}, CS_{n-k}\}, \{UC, CS_{k'-1}\}, \{UCS_{k-1}\})$ を実現しており、 $k' < k$ の場合、 $(n, DS, LS1, LS2) = (n, \{UCS_{n-k}, CS_{n-k}\}, \{UC, CS_{k'-1}\}, \{UCS_{k-1}, CS_{k-1}\})$ を実現している。

さらに、ユーザの所有物に記録してあるデータと記憶情報のすべて、あるいはそれらの一部も秘密分散器で (k', n) 分散データに分割して各サーバにも保存しておけば、ユーザが仮に記憶情報を忘れたとしてもオフライン解析により (所有物の記録情報と) 記憶情報を復元できる。その際、1) すべてを分散する場合はオフライン解析を省くことができ、2) 一部を分散する場合はその量に応じてオフライン解析の計算量を削減することができる。この機能により、ユーザは第三者にデータを復号する権限を与える際に、データを復元する

容易さ（攻撃者が $\{CS_k'\}$ を得た場合のデータ復元の容易さとも同じ）を調整できるようになる。

＜端末にデータを保存しない場合の遠隔分散保存装置におけるデータ復元処理＞

次に、図 2 3 を参照して、端末にデータを保存しない場合の遠隔分散保存装置におけるデータ復元処理について説明する。図 2 3 は、端末にデータを保存しない場合の遠隔分散保存装置 5 の構成を示すブロック図である。

データ復元器 5 4 は、入力された $DI D$ に対応する分散データ $S' 1, \dots, S' n$ の内少なくとも k 個を通信器 5 2 を介してサーバ $ID 1, ID 2, \dots, ID n$ から受け取る。データ復元器 5 4 は $S' 1, \dots, S' n$ の内少なくとも k 個を処理して $DAT A$ を復元する。同様の方法で、保存データの一覧を復元することも可能である。また、ユーザの端末装置 2 1（認証データ更新モード）を適当な間隔（記憶情報がオフライン全数探索で求まる間隔より短い間隔で、例えば、認証を行う度や 2, 3 日に 1 回など）で動作させ、 P' と各サーバに記録されている H を更新する情報 $UP', UH 1, \dots, UH n$ を生成させ、それらを更新する。

次に、図 2 4 を参照して、図 2 3 に示すデータ復元器 5 4 の構成を説明する。調整器 5 4 1 は、入力された n 台のサーバ $ID, ID 1, ID 2, \dots, ID n$ と $DI D$ を出力する。また、調整器 5 4 1 は $DI D$ からデータ伸長器 5 4 2 の入力 x を生成しデータ伸長器 5 4 2 へ渡す。データ伸長器 5 4 2 は対応する情報 H を出力し、それを暗号器 5 4 3 に渡す。暗号器 5 4 3 は得られた分散データ $S' 1, S' 2, \dots, S' n$ の内、暗号化されているものを復号し $S 1, S 2, \dots, S n$ を秘密分散復元器に渡す。秘密分散復元器 5 4 4 は渡された分散データから $DAT A$ を復元する。なお、改ざん検出を行い改ざんの行われていない分散データのみを k 個渡してもよい。

＜端末にも分散データを保存する場合の遠隔分散保存装置におけるデータ保

存処理＞

次に、図 2 5 を参照して、端末にも分散データを保存する場合の遠隔分散保存装置におけるデータ保存処理について説明する。図 2 5 は、端末にも分散データを保存する場合の遠隔分散保存装置 5 の構成を示すブロック図である。ここでは、図 2 1 に示す構成を異なる部分のみを説明する。

ユーザは分散保存したいデータ DATA を自分の端末装置 2 1 で処理し、手元に残すデータ DL と n 台のサーバに保存するデータ RS' 1, ..., RS' n へと分割する。DL は手元の記録装置 5 5 に保存され、分割データ RS' 1 はデータの ID である DID と共にユーザの端末装置 2 1 によりサーバの認証装置と共有された鍵 SK i を用いて作成された安全な通信路を通してサーバ ID i へと送られ保存される。同様の方法で、保存データの一覧情報を分割し、サーバに保存することも可能である。

このように、分散データの一部を自分の端末装置 2 1 に置くことで、サーバとの通信量を削減できる。破損したサーバの組み合わせに応じて通信量が変わる方式と変わらない方式が可能であるが、変わる方法の場合 n 台のサーバ全体の記憶領域を削減できる。なお、漏洩と破損に対する耐性は、データを手元に置かない場合と同等に保つことができる。

次に、図 2 6 を参照して、図 2 5 に示すデータ分割器 5 1 の構成を説明する。ここでは、図 2 2 に示す構成と異なる部分のみを説明する。暗号器 5 1 5 は乱数生成器 5 1 6 から乱数 R を受け取り、これを鍵として保存データを暗号化し、それを DL として出力する。調整器 5 1 1 は入力されたパラメータ n, k を秘密分散器 5 1 2 に渡す。秘密分散器 5 1 2 は入力されたパラメータ n, k に従い R を (k, n) 分散データ RS 1, RS 2, ..., RS n に変換する。次に、調整器 5 1 1 はデータの ID である DID からデータ伸長器 5 1 3 の入力 x を生成しデータ伸長器 5 1 3 へ渡す。データ伸長器 5 1 3 は対応する情報 H を出力し、これを鍵として暗号器 5 1 4 に渡す。ここでの H はオフラインでの全数探索に強い長さのものを利用する。データ伸長器 5 1 3 が短い H を出力する場合には異なる x を複数のデータ伸長器 5 1 3 に渡し、得られた複数の H を利

用する。暗号器 5 1 4 は、H を鍵として $n - k + 1$ 個以上の分散データを暗号化する。また、 RS_1, \dots, RS_{n-k+1} のそれぞれに改ざん検出符号を付けてもよい。暗号器の出力が RS'_1, \dots, RS'_n である。

なお、 (k, n) 分散データ以外にも任意のアクセス構造を持った分散データを利用することが可能である。また、秘密分散器 5 1 2 には多項式や行列を使った情報量的に安全な分散方式や、暗号を使う計算量的に安全な分散方式を用いることが可能であるが、R のサイズが小さい場合には、計算量的に安全な分散方式を用いることによるサイズの削減効果は小さいため、情報量的に安全な分散方式を用いた方がよい。

< 端末にも分散データを保存する場合の遠隔分散保存装置におけるデータ復元処理 >

次に、図 2 7 を参照して、端末にも分散データを保存する場合の遠隔分散保存装置におけるデータ復元処理について説明する。図 2 7 は、端末にも分散データを保存する場合の遠隔分散保存装置 5 の構成を示すブロック図である。ここでは、図 2 3 に示す構成と異なる部分のみを説明する。

データ復元器 5 4 は入力された DID に対応する分散データ RS'_1, \dots, RS'_n の内少なくとも k 個を通信器を介してサーバ ID 1, ID 2, \dots , ID n から受け取る。データ復元器 5 4 は RS'_1, \dots, RS'_n の内少なくとも k 個を処理して DATA を復元する。同様の方法で、保存データの一覧を復元することも可能である。

次に、図 2 8 を参照して、図 2 7 に示すデータ復元器 5 4 の構成を説明する。ここでは、図 2 4 に示す構成と異なる部分のみを説明する。調整器 5 4 1 は、DID からデータ伸長器 5 4 2 の入力 x を生成しデータ伸長器 5 4 2 へ渡す。データ伸長器 5 4 2 は対応する情報 H を出力し、それを復号器 5 4 3 に渡す。復号器 5 4 3 は得られた分散データ $RS'_1, RS'_2, \dots, RS'_n$ の内、暗号化されているものを復号し RS_1, \dots, RS_n を秘密分散復元器 5 4 4 に渡す。秘密分散復元器 5 4 4 は渡された分散データから DATA を復号器 5 4 5

によって復元する。また、改ざん検出を行い改ざんの行われていない分散データを k 個渡してもよい。

なお、図面に示す各処理部の機能を実現するためのプログラムをコンピュータ読み取り可能な記録媒体に記録して、この記録媒体に記録されたプログラムをコンピュータシステムに読み込ませ、実行することにより認証処理、鍵交換処理を行ってもよい。なお、ここでいう「コンピュータシステム」とは、OSや周辺機器等のハードウェアを含むものとする。また、「コンピュータシステム」は、ホームページ提供環境（あるいは表示環境）を備えたWWWシステムも含むものとする。また、「コンピュータ読み取り可能な記録媒体」とは、フレキシブルディスク、光磁気ディスク、ROM、CD-ROM等の可搬媒体、コンピュータシステムに内蔵されるハードディスク等の記憶装置のことをいう。さらに「コンピュータ読み取り可能な記録媒体」とは、インターネット等のネットワークや電話回線等の通信回線を介してプログラムが送信された場合のサーバやクライアントとなるコンピュータシステム内部の揮発性メモリ（RAM）のように、一定時間プログラムを保持しているものも含むものとする。

また、上記プログラムは、このプログラムを記憶装置等に格納したコンピュータシステムから、伝送媒体を介して、あるいは、伝送媒体中の伝送波により他のコンピュータシステムに伝送されてもよい。ここで、プログラムを伝送する「伝送媒体」は、インターネット等のネットワーク（通信網）や電話回線等の通信回線（通信線）のように情報を伝送する機能を有する媒体のことをいう。また、上記プログラムは、前述した機能の一部を実現するためのものであってもよい。さらに、前述した機能をコンピュータシステムにすでに記録されているプログラムとの組み合わせで実現できるもの、いわゆる差分ファイル（差分プログラム）であってもよい。

この発明によれば、端末装置側あるいはサーバ側から装置内に保存している情報が漏れたとしてもオフライン解析によってパスワードを見つけだすことができないため、サーバの不正利用を防止することが可能になるという効果が得られる。また、装置内に保存されている情報を盗まれないようにするための耐タンパー性のモジュールを使用する必要がないため、装置構成を簡単にすることができる。また、公開鍵暗号システムのように複雑な鍵管理処理を行う必要がないため、計算処理を向上させることができるとともに、処理内容を簡単にすることができる。また、複数のサーバに対しても拡張することができる。

さらに、各サーバと端末装置とでユーザIDを同期させながら動的に変化させることで、盗聴者がユーザIDを用いてユーザのプライバシー情報を紐付けすることを防止することができる。

請 求 の 範 囲

1. 端末装置とサーバ間において相互に認証を行う認証システムであって、
前記端末装置は、
ユーザ保存用の認証情報 P' を予め記憶しておく記憶手段と、
前記記憶手段から読み出した認証情報 P' と認証時に入力されたパスワード
を入力として所定の計算式により値 P を求める結合手段と、
前記値 P と内部において発生させた乱数を入力として所定の計算式により
値 Y_1 を求め、前記サーバへ送信するマスク演算手段と、
前記値 P と内部において発生させた乱数と前記サーバから受信した値 Y_2
を入力として所定の計算式により値 MK を求めるマスター鍵生成手段と
を備え、
前記サーバは、
サーバ登録用のパスワード認証データ H を予め記憶しておく記憶手段と、
前記記憶手段から読み出したパスワード認証データ H と内部において発生
させた乱数を入力として所定の計算式により値 Y_2 を求め、前記端末装置へ送
信するマスク演算手段と、
前記パスワード認証データ H と内部において発生させた乱数と前記端末装
置から受信した値 Y_1 を入力として所定の計算式により値 MK を求めるマス
ター鍵生成手段と
を備えたことを特徴とする認証システム。
2. ユーザが予め決定したパスワードに基づいて、前記パスワード認証デー
タ H と前記認証情報 P' を求めるデータ伸長手段をさらに備えたことを特徴
とする請求項 1 に記載の認証システム。
3. 前記端末装置は、
前記値 MK を入力として所定の計算式により値 V_1 を求め、サーバへ送信す
るとともに、前記サーバから受信した値 V_2 と前記値 MK を入力として所定の
計算式による値 V_2 と比較照合し、一致した場合にサーバを認証する認証結果

判断手段をさらに備え、

前記サーバは、

前記値MKを入力として所定の計算式により値V2を求め、端末装置へ送信するとともに、前記端末装置から受信した値V1と前記値MKを入力として所定の計算式による値V1と比較照合し、一致した場合に端末装置を認証する認証結果判断手段をさらに備えた

ことを特徴とする請求項1または2に記載の認証システム。

4. 前記端末装置及びサーバは、相互の認証が行われた場合に、セッション鍵を生成するセッション鍵生成手段をそれぞれに備えたことを特徴とする請求項3に記載の認証システム。

5. 前記認証情報P'は、多項式であることを特徴とする請求項1から4のいずれかに記載の認証システム。

6. 前記認証情報P'は、多項式とハッシュ関数であることを特徴とする請求項1から4のいずれかに記載の認証システム。

7. 前記認証情報P'は、ハッシュ関数であることを特徴とする請求項1から4のいずれかに記載の認証システム。

8. 前記認証情報P'は、疑似乱数関数であることを特徴とする請求項1から4のいずれかに記載の認証システム。

9. 端末装置とサーバ間において相互に認証を行う認証システムにおける端末装置上で動作する認証プログラムであって、

ユーザ保存用の認証情報P'を予め記憶しておく記憶処理と、

前記記憶しておいた認証情報P'と認証時に入力されたパスワードを入力として所定の計算式により値Pを求める結合処理と、

前記値 P と内部において発生させた乱数を入力として所定の計算式により値 Y 1 を求め、前記サーバへ送信するマスク演算処理と、

前記値 P と内部において発生させた乱数と前記サーバから受信した値 Y 2 を入力として所定の計算式により値 MK を求めるマスター鍵生成処理とをコンピュータに行わせることを特徴とする認証プログラム。

10. ユーザが予め決定したパスワードに基づいて、前記認証情報 P' を求めるデータ伸長処理をさらにコンピュータに行わせることを特徴とする請求項 9 に記載の認証プログラム。

11. 前記値 MK を入力として所定の計算式により値 V 1 を求め、サーバへ送信するとともに、前記サーバから受信した値 V 2 と前記値 MK を入力として所定の計算式による値 V 2 と比較照合し、一致した場合にサーバを認証する認証結果判断処理をさらにコンピュータに行わせることを特徴とする請求項 9 または 10 に記載の認証プログラム。

12. 端末装置とサーバ間において相互に認証を行う認証システムにおけるサーバ上で動作する認証プログラムであって、

サーバ登録用のパスワード認証データ H を予め記憶しておく記憶処理と、

前記記憶しておいたパスワード認証データ H と内部において発生させた乱数を入力として所定の計算式により値 Y 2 を求め、前記端末装置へ送信するマスク演算処理と、

前記パスワード認証データ H と内部において発生させた乱数と前記端末装置から受信した値 Y 1 を入力として所定の計算式により値 MK を求めるマスター鍵生成処理と

をコンピュータに行わせることを特徴とする認証プログラム。

13. ユーザが予め決定したパスワードに基づいて、前記パスワード認証データ H を求めるデータ伸長処理をさらにコンピュータに行わせることを特徴

とする請求項 1 2 に記載の認証プログラム。

1 4. 前記値MKを入力として所定の計算式により値V 2 を求め、端末装置へ送信するとともに、前記端末装置から受信した値V 1 と前記値MKを入力として所定の計算式による値V 1 と比較照合し、一致した場合に端末装置を認証する認証結果判断処理をさらにコンピュータに行わせることを特徴とする請求項 1 2 または 1 3 に記載の認証プログラム。

1 5. 端末装置及びサーバにおいて相互の認証が行われた場合に、セッション鍵を生成するセッション鍵生成処理をそれぞれに備えたことを特徴とする請求項 1 1 または 1 4 に記載の認証プログラム。

1 6. 前記認証情報P' は、多項式であることを特徴とする請求項 9 から 1 5 のいずれかに記載の認証プログラム。

1 7. 前記認証情報P' は、多項式とハッシュ関数であることを特徴とする請求項 9 から 1 5 のいずれかに記載の認証プログラム。

1 8. 前記認証情報P' は、ハッシュ関数であることを特徴とする請求項 9 から 1 5 のいずれかに記載の認証プログラム。

1 9. 前記認証情報P' は、疑似乱数関数であることを特徴とする請求項 9 から 1 5 のいずれかに記載の認証プログラム。

2 0. 前記端末装置は、

更新情報T' を発生させる発生手段と、

前記記憶手段に記憶されている認証情報P' と前記更新情報T' を入力し、所定の計算式によってサーバ更新用のパスワード認証データH' と新たな認証情報P' を求め、サーバ更新用のパスワード認証データH' を前記サーバへ送

信するとともに、新たな認証情報 P' を前記記憶手段に記憶する更新情報生成手段と

を備え、

前記サーバは、

前記端末装置から送信されたサーバ更新用のパスワード認証データ H' と前記記憶手段に記憶されたパスワード認証データ H を入力し、所定の計算式によって新たなパスワード認証データ H を求めて前記記憶手段に記憶されているパスワード認証データ H を更新する更新情報生成手段を備えた

ことを特徴とする請求項 2 に記載の認証システム。

2 1. 前記端末装置は、

秘密情報 S' を発生させる発生手段と、

前記記憶手段に記憶されている認証情報 P' と前記秘密情報 S' と新しいパスワードを入力し、所定の計算式によってサーバ更新用のパスワード認証データ H' と新たな認証情報 P' を求め、サーバ更新用のパスワード認証データ H' を前記サーバへ送信するとともに、新たな認証情報 P' を前記記憶手段に記憶する更新情報生成手段と

を備え、

前記サーバは、

前記端末装置から送信されたサーバ更新用のパスワード認証データ H' と前記記憶手段に記憶されたパスワード認証データ H を入力し、所定の計算式によって新たなパスワード認証データ H を求めて前記記憶手段に記憶されているパスワード認証データ H を更新する更新情報生成手段を備えた

ことを特徴とする請求項 2 に記載の認証システム。

2 2. 端末装置とサーバ間において相互に認証を行う認証システムであって、前記端末装置は、

ユーザ保存用の認証情報 P' と RSA 公開鍵 (N , e) を予め記憶しておく記憶手段と、

前記記憶手段から読み出した認証情報 P' と認証時に入力されたパスワードを入力として所定の計算式により値 W を求める結合手段と、

前記値 W と前記記憶手段から読み出した RSA 公開鍵 (N, e) と内部において発生させた乱数 T を入力として所定の計算式により値 Z を求め、前記サーバへ送信するマスク演算手段と

を備え、

前記サーバは、

サーバ登録用のパスワード認証データ H と RSA 秘密鍵 (N, d) を予め記憶しておく記憶手段と、

前記記憶手段から読み出したパスワード認証データ H と RSA 秘密鍵 (N, d) と前記端末装置から受信した値 Z を入力として所定の計算式により値 T を求めるマスター鍵生成手段と

を備えたことを特徴とする認証システム。

23. ユーザが予め決定したパスワードに基づいて、前記パスワード認証データ H と前記認証情報 P' を求めるデータ伸長手段をさらに備えたことを特徴とする請求項 22 に記載の認証システム。

24. 前記 RSA 公開鍵 (N, e) と前記 RSA 秘密鍵 (N, d) を求める RSA 鍵生成手段をさらに備えたことを特徴とする請求項 22 に記載の認証システム。

25. 前記端末装置は、

前記サーバから受信した値 V_2 と前記乱数 T を入力として所定の計算式による値 V_2 と比較照合し、一致した場合にサーバを認証する認証結果判断手段と、

前記乱数 T を入力として所定の計算式により値 V_1 を求め、前記サーバへ送信する検証子生成手段をさらに備え、

前記サーバは、

前記値 T を入力として所定の計算式により値 V_2 を求め、前記端末装置へ送信する検証子生成手段と、

前記端末装置から受信した値 V_1 と前記値 T を入力として所定の計算式による値 V_1 と比較照合し、一致した場合に端末装置を認証する認証結果判断手段をさらに備えた

ことを特徴とする請求項 22、23 または 24 に記載の認証システム。

26. 前記端末装置及びサーバは、相互の認証が行われた場合に、セッション鍵を生成するセッション鍵生成手段をそれぞれに備えたことを特徴とする請求項 25 に記載の認証システム。

27. 前記認証情報 P' は、多項式と FDH 関数であることを特徴とする請求項 22 から 26 のいずれかに記載の認証システム。

28. 前記認証情報 P' は、 FDH 関数であることを特徴とする請求項 22 から 26 のいずれかに記載の認証システム。

29. 前記 RSA 公開鍵 (N, e) は、安全な通信を用いることを特徴とする請求項 22 から 26 のいずれかに記載の認証システム。

30. 前記 RSA 公開鍵 (N, e) は、安全ではない通信を用いることを特徴とする請求項 22 から 26 のいずれかに記載の認証システム。

31. 端末装置とサーバ間において相互に認証を行う認証システムにおける端末装置上で動作する認証プログラムであって、

ユーザ保存用の認証情報 P' と RSA 公開鍵 (N, e) を予め記憶しておく記憶処理と、

前記記憶しておいた認証情報 P' と認証時に入力されたパスワードを入力として所定の計算式により値 W を求める結合処理と、

前記値 W と前記記憶しておいた $RS A$ 公開鍵 (N, e) と内部において発生させた乱数 T を入力として所定の計算式により値 Z を求め、前記サーバへ送信するマスク演算処理と

をコンピュータに行わせることを特徴とする認証プログラム。

32. ユーザが予め決定したパスワードに基づいて、前記認証情報 P' を求めるデータ伸長処理をさらにコンピュータに行わせることを特徴とする請求項31に記載の認証プログラム。

33. 前記 $RS A$ 公開鍵 (N, e) を求める $RS A$ 鍵生成処理をさらにコンピュータに行わせることを特徴とする請求項31に記載の認証プログラム。

34. 前記サーバから受信した値 $V2$ と前記乱数 T を入力として所定の計算式による値 $V2$ と比較照合し、一致した場合にサーバを認証する認証結果判断処理と、

前記乱数 T を入力として所定の計算式により値 $V1$ を求め、前記サーバへ送信する検証子生成処理をさらにコンピュータに行わせることを特徴とする請求項31、32または33に記載の認証プログラム。

35. 端末装置とサーバ間において相互に認証を行う認証システムにおけるサーバ上で動作する認証プログラムであって、

サーバ登録用のパスワード認証データ H と $RS A$ 秘密鍵 (N, d) を予め記憶しておく記憶処理と、

前記記憶しておいたパスワード認証データ H と $RS A$ 秘密鍵 (N, d) と前記端末装置から受信した値 Z を入力として所定の計算式により値 T を求めるマスター鍵生成処理と

をコンピュータに行わせることを特徴とする認証プログラム。

36. ユーザが予め決定したパスワードに基づいて、前記パスワード認証デ

ータHを求めるデータ伸長処理をさらにコンピュータに行わせることを特徴とする請求項35に記載の認証プログラム。

37. 前記RSA秘密鍵(N, d)を求めるRSA鍵生成処理をさらにコンピュータに行わせることを特徴とする請求項35に記載の認証プログラム。

38. 前記値Tを入力として所定の計算式により値V2を求め、前記端末装置へ送信する検証子生成処理と、

前記端末装置から受信した値V1と前記値Tを入力として所定の計算式による値V1と比較照合し、一致した場合に端末装置を認証する認証結果判断処理をさらにコンピュータに行わせることを特徴とする請求項35、36または37に記載の認証プログラム。

39. 端末装置及びサーバにおいて相互の認証が行われた場合に、セッション鍵を生成するセッション鍵生成処理をそれぞれに備えたことを特徴とする請求項34または38に記載の認証プログラム。

40. 前記認証情報P'は、多項式とFDH関数であることを特徴とする請求項31から39のいずれかに記載の認証プログラム。

41. 前記認証情報P'は、FDH関数であることを特徴とする請求項31から39のいずれかに記載の認証プログラム。

42. 前記RSA公開鍵(N, e)は、安全な通信を用いることを特徴とする請求項31から39のいずれかに記載の認証プログラム。

43. 前記RSA公開鍵(N, e)は、安全ではない通信を用いることを特徴とする請求項31から39のいずれかに記載の認証プログラム。

4 4. 前記端末装置は、

更新情報 T' を発生させる発生手段と、

前記記憶手段に記憶されている認証情報 P' と前記更新情報 T' を入力し、所定の計算式によってサーバ更新用のパスワード認証データ H' と新たな認証情報 P' を求め、サーバ更新用のパスワード認証データ H' を前記サーバへ送信するとともに、新たな認証情報 P' を前記記憶手段に記憶する更新情報生成手段と

を備え、

前記サーバは、

前記端末装置から送信されたサーバ更新用のパスワード認証データ H' と前記記憶手段に記憶されたパスワード認証データ H を入力し、所定の計算式によって新たなパスワード認証データ H を求めて前記記憶手段に記憶されているパスワード認証データ H を更新する更新情報生成手段を備えた

ことを特徴とする請求項 2 3 に記載の認証システム。

4 5. 前記端末装置は、

前記記憶手段に記憶されている認証情報 P' と前記乱数 T を入力し、所定の計算式によって新たな認証情報 P' を求め、新たな認証情報 P' を前記記憶手段に記憶する更新情報生成手段と

を備え、

前記サーバは、

前記記憶手段に記憶されたパスワード認証データ H と前記マスター鍵生成手段によって求めた値 T を入力し、所定の計算式によって新たなパスワード認証データ H を求めて前記記憶手段に記憶されているパスワード認証データ H を更新する更新情報生成手段を備えた

ことを特徴とする請求項 2 2 に記載の認証システム。

4 6. 前記端末装置は、

秘密情報 S' を発生させる発生手段と、

前記記憶手段に記憶されている認証情報 P' と前記秘密情報 S' と新しいパスワードを入力し、所定の計算式によってサーバ更新用のパスワード認証データ H' と新たな認証情報 P' を求め、サーバ更新用のパスワード認証データ H' を前記サーバへ送信するとともに、新たな認証情報 P' を前記記憶手段に記憶する更新情報生成手段と

を備え、

前記サーバは、

前記端末装置から送信されたサーバ更新用のパスワード認証データ H' と前記記憶手段に記憶されたパスワード認証データ H を入力し、所定の計算式によって新たなパスワード認証データ H を求めて前記記憶手段に記憶されているパスワード認証データ H を更新する更新情報生成手段を備えた

ことを特徴とする請求項 2 3 に記載の認証システム。

47. 端末装置と複数のサーバ間において相互に認証を行い、前記端末装置内の保存対象のデータを前記サーバ内に分散して保存する遠隔分散保存システムであって、

前記端末装置は、

ユーザが予め決定しておいたパスワードに基づいて、サーバ登録用のパスワード認証データ H とユーザ保存用の認証情報 P' を求めるデータ伸長手段と、

前記データ伸長手段によって求めた認証情報 P' を予め記憶しておく記憶手段と、

前記記憶手段から読み出した認証情報 P' と認証時に入力されたパスワードを入力として所定の計算式により値 P を求める結合手段と、

前記値 P と内部において発生させた乱数を入力として所定の計算式により値 Y_1 を求め、前記サーバへ送信するマスク演算手段と、

前記値 P と内部において発生させた乱数と前記サーバから受信した値 Y_2 を入力として所定の計算式により値 MK を求めるマスター鍵生成手段と、

前記値 MK を入力として所定の計算式により値 V_1 を求め、サーバへ送信するとともに、前記サーバから受信した値 V_2 と値 V_1 と比較照合し、一致した場合

にサーバを認証する認証結果判断手段と、

サーバの認証が行われた場合に、セッション鍵 SK をサーバの数だけ生成するセッション鍵生成手段と、

前記保存対象のデータを分割して、認証したサーバの数と同数の分割データを得るデータ分割手段と、

前記分割データのそれぞれと保存対象のデータを識別する識別情報とを、保存先のサーバと共有した前記セッション鍵 SK を用いて暗号化して、各サーバに対して送信するデータ保存手段と、

分割データが保存された各サーバから分割データを受信し、前記保存対象のデータを復元するデータ復元手段と

を備え、

前記サーバは、

前記データ伸長手段によって求めたパスワード認証データ H を予め記憶しておく記憶手段と、

前記記憶手段から読み出したパスワード認証データ H と内部において発生させた乱数を入力として所定の計算式により値 Y_2 を求め、前記端末装置へ送信するマスク演算手段と、

前記パスワード認証データ H と内部において発生させた乱数と前記端末装置から受信した値 Y_1 を入力として所定の計算式により値 MK を求めるマスター鍵生成手段と、

前記値 MK を入力として所定の計算式により値 V_2 を求め、端末装置へ送信するとともに、前記端末装置から受信した値 V_1 と値 V_2 と比較照合し、一致した場合に端末装置を認証する認証結果判断手段と、

端末装置の認証が行われた場合に、セッション鍵を生成するセッション鍵生成手段と、

端末装置から受信した分割データを受信するデータ受信手段と、

前記分割データを記憶するデータ記憶手段と、

前記データ記憶手段に保存されている分割データを読み出して端末装置へ送信するデータ送信手段と

を備えたことを特徴とする遠隔分散保存システム。

48. 前記分割データの一部を前記端末装置内に保存することを特徴とする請求項47に記載の遠隔分散保存システム。

49. 端末装置と複数のサーバ間において相互に認証を行い、前記端末装置内の保存対象のデータを前記サーバ内に分散して保存する遠隔分散保存システムにおける端末装置上で動作する遠隔分散保存プログラムであって、

ユーザが予め決定しておいたパスワードに基づいて、サーバ登録用のパスワード認証データHとユーザ保存用の認証情報P'を求めるデータ伸長処理と、

前記データ伸長処理によって求めた認証情報P'を予め記憶しておく記憶処理と、

前記記憶処理から読み出した認証情報P'と認証時に入力されたパスワードを入力として所定の計算式により値Pを求める結合処理と、

前記値Pと内部において発生させた乱数を入力として所定の計算式により値Y1を求め、前記サーバへ送信するマスク演算処理と、

前記値Pと内部において発生させた乱数と前記サーバから受信した値Y2を入力として所定の計算式により値MKを求めるマスター鍵生成処理と、

前記値MKを入力として所定の計算式により値V1を求め、サーバへ送信するとともに、前記サーバから受信した値V2と値V1と比較照合し、一致した場合にサーバを認証する認証結果判断処理と、

サーバの認証が行われた場合に、セッション鍵SKをサーバの数だけ生成するセッション鍵生成処理と、

前記保存対象のデータを分割して、認証したサーバの数と同数の分割データを得るデータ分割処理と、

前記分割データのそれぞれと保存対象のデータを識別する識別情報とを、保存先のサーバと共有した前記セッション鍵SKを用いて暗号化して、各サーバに対して送信するデータ保存処理と、

分割データが保存された各サーバから分割データを受信し、前記保存対象のデ

ータを復元するデータ復元処理と

をコンピュータに行わせることを特徴とする遠隔分散保存プログラム

50. 端末装置と複数のサーバ間において相互に認証を行い、前記端末装置内の保存対象のデータを前記サーバ内に分散して保存する遠隔分散保存システムにおけるサーバ上で動作する遠隔分散保存プログラムであって、

データ伸長処理によって求めたパスワード認証データHを予め記憶しておく記憶処理と、

前記記憶処理から読み出したパスワード認証データHと内部において発生させた乱数を入力として所定の計算式により値Y2を求め、前記端末装置へ送信するマスク演算処理と、

前記パスワード認証データHと内部において発生させた乱数と前記端末装置から受信した値Y1を入力として所定の計算式により値MKを求めるマスター鍵生成処理と、

前記値MKを入力として所定の計算式により値V2を求め、端末装置へ送信するとともに、前記端末装置から受信した値V1と値V2と比較照合し、一致した場合に端末装置を認証する認証結果判断処理と、

端末装置の認証が行われた場合に、セッション鍵を生成するセッション鍵生成処理と、

端末装置から受信した分割データを受信するデータ受信処理と、

前記分割データを記憶するデータ記憶処理と、

前記データ記憶処理に保存されている分割データを読み出して端末装置へ送信するデータ送信処理と

をコンピュータに行わせることを特徴とする遠隔分散保存プログラム。

1/22

図1

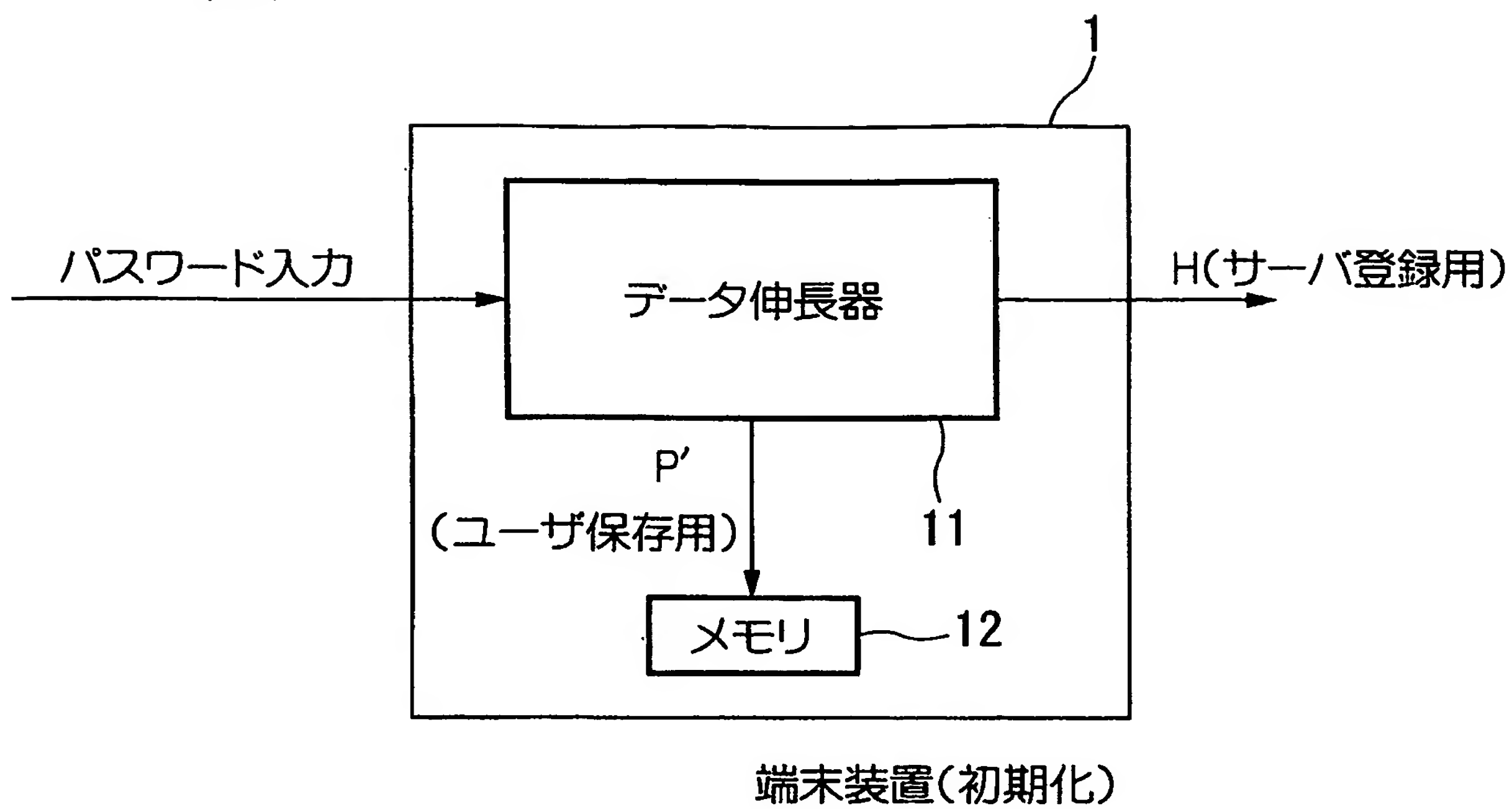
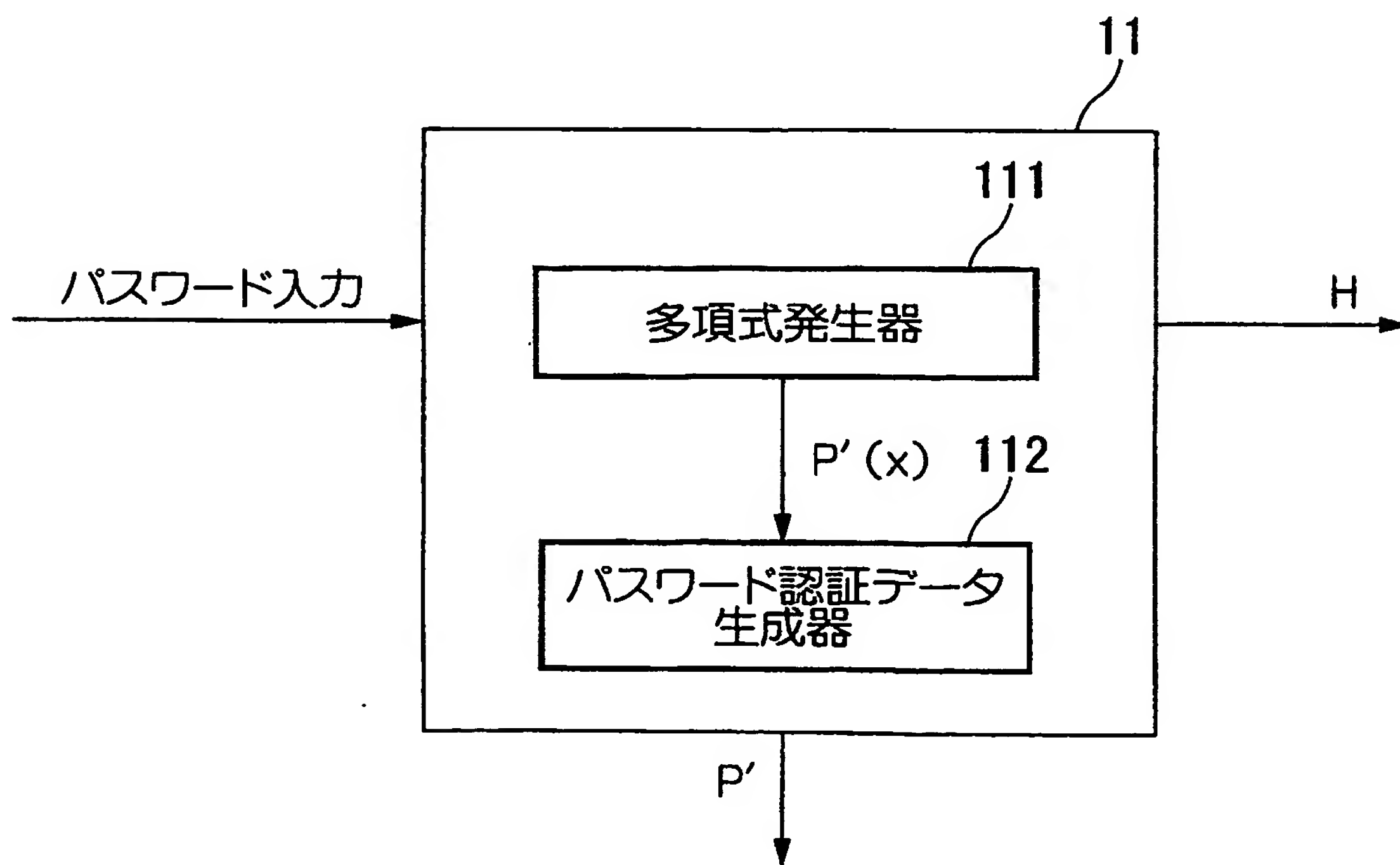


図2



2/22

図3

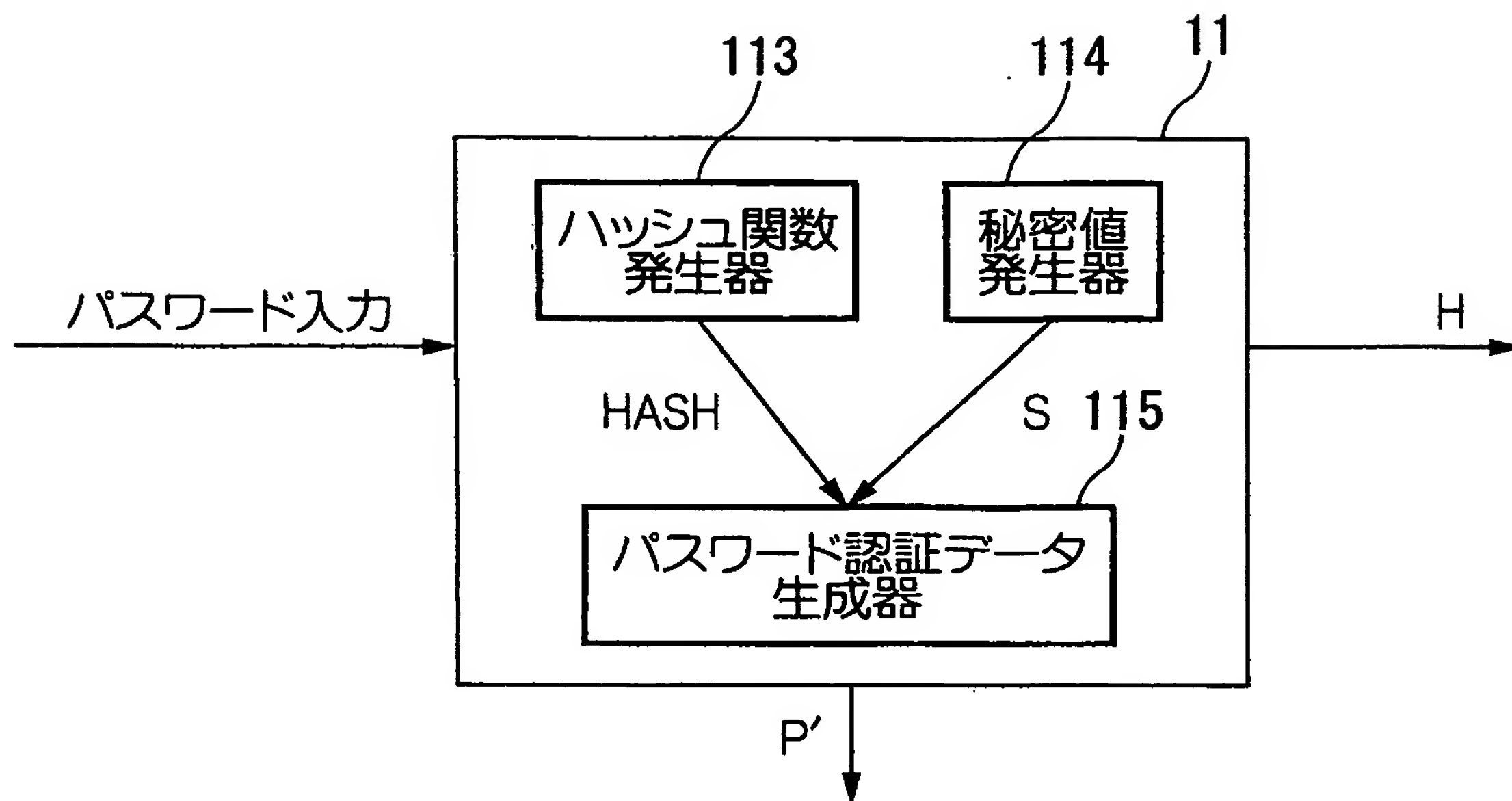
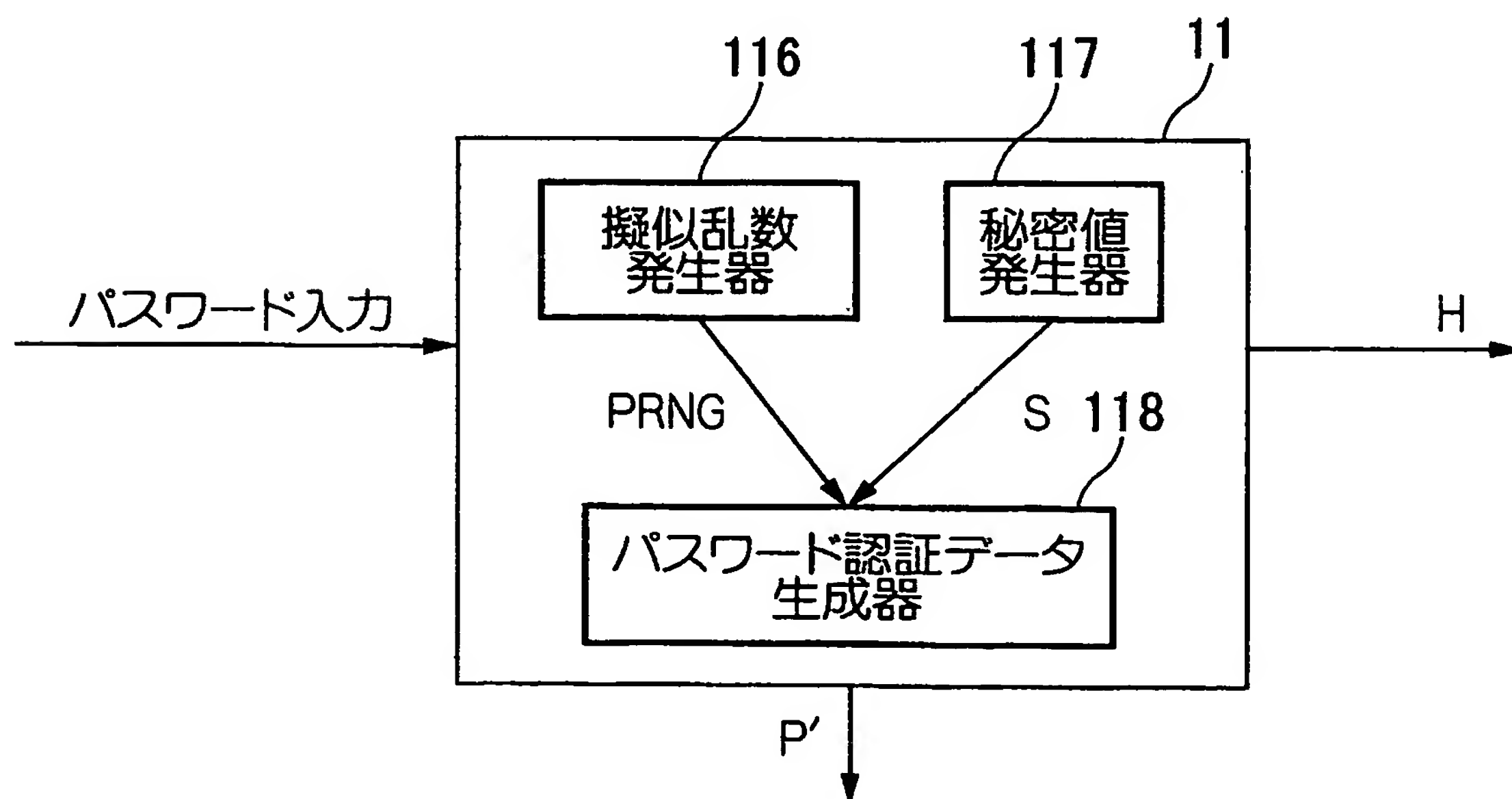
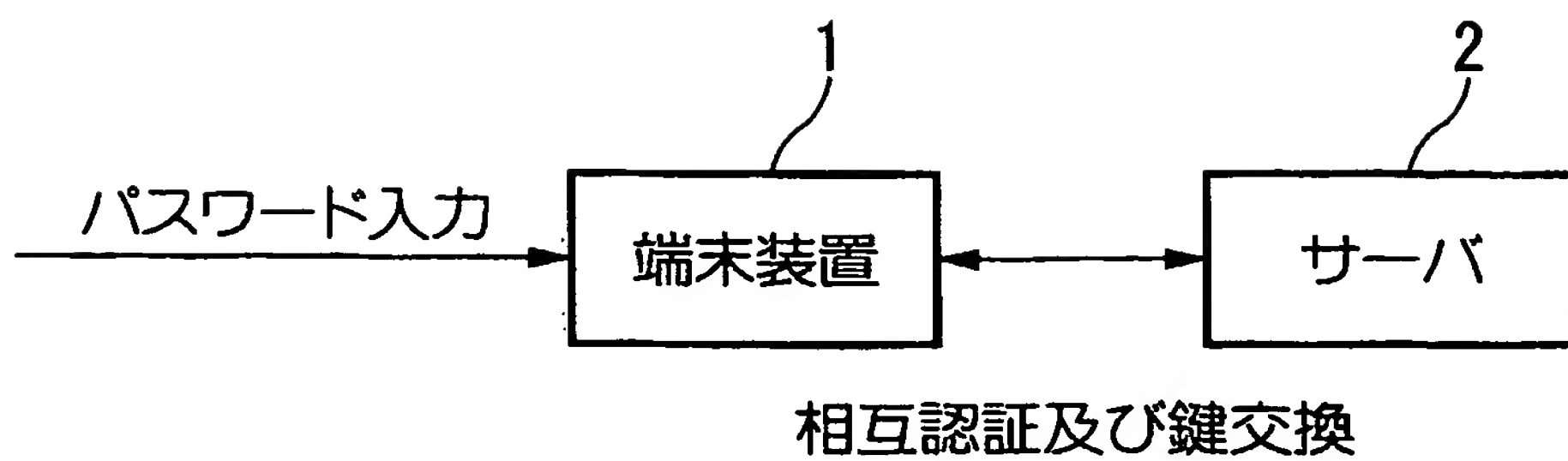


図4

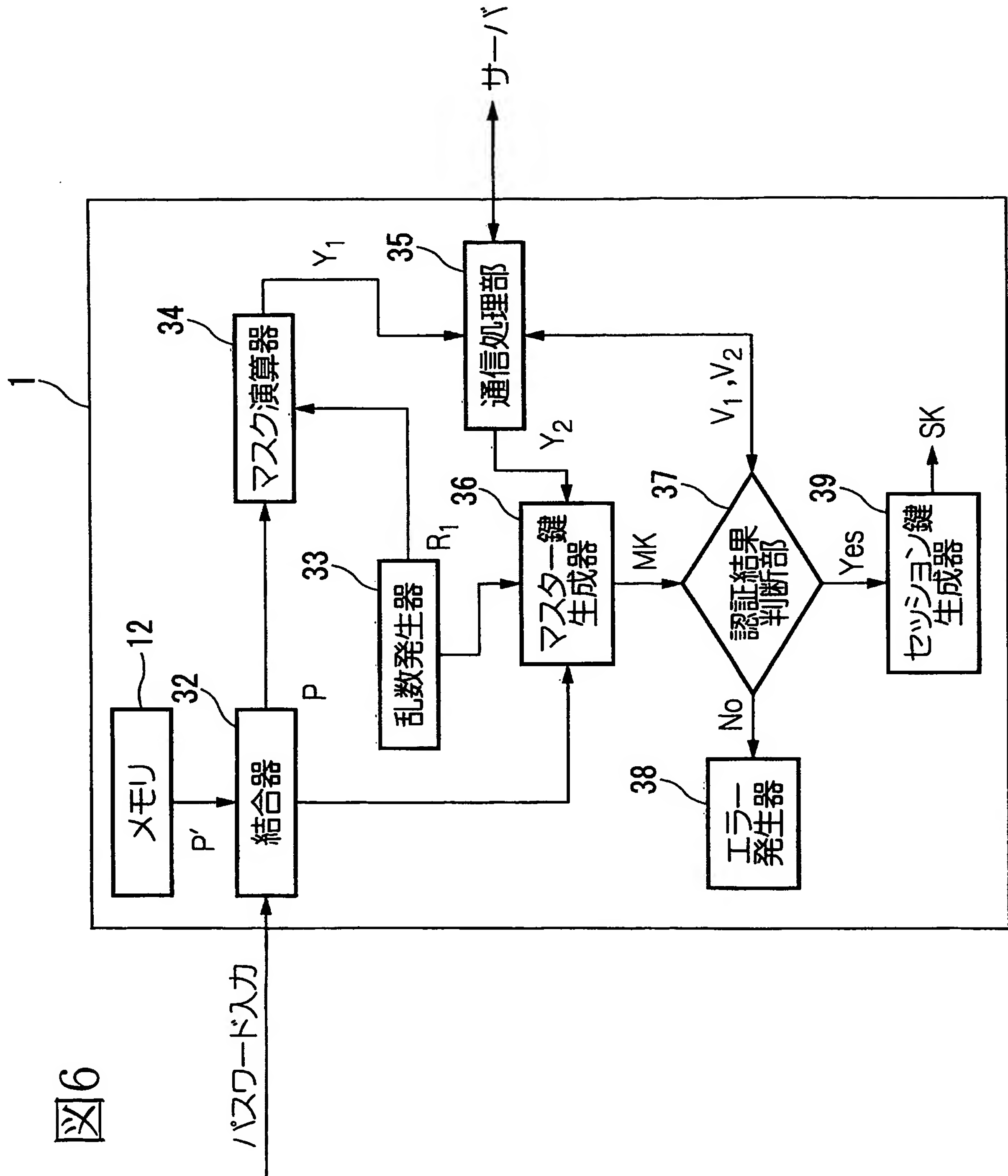


3/22

図5

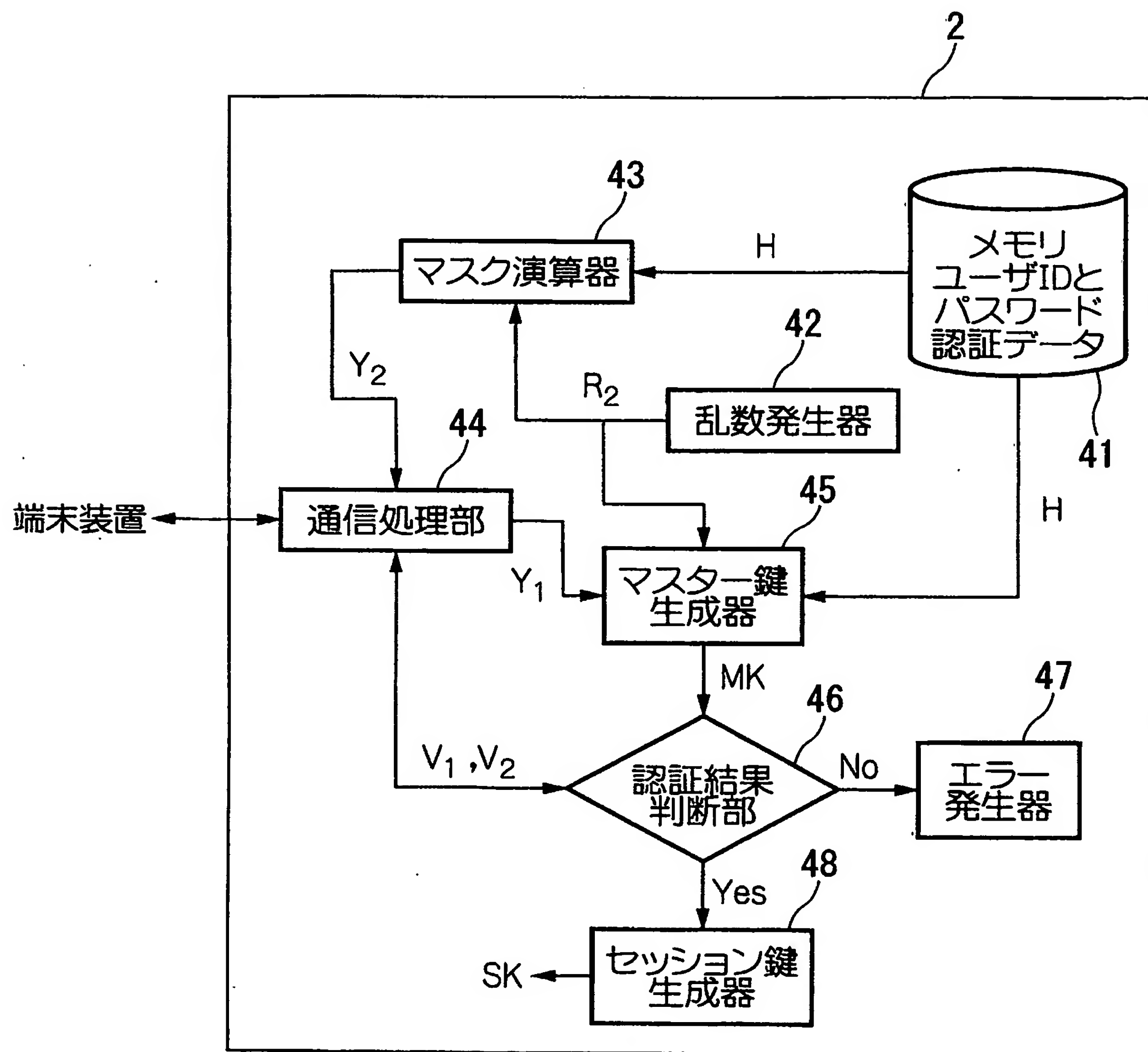


4/22



5/22

図7



6/22

図8

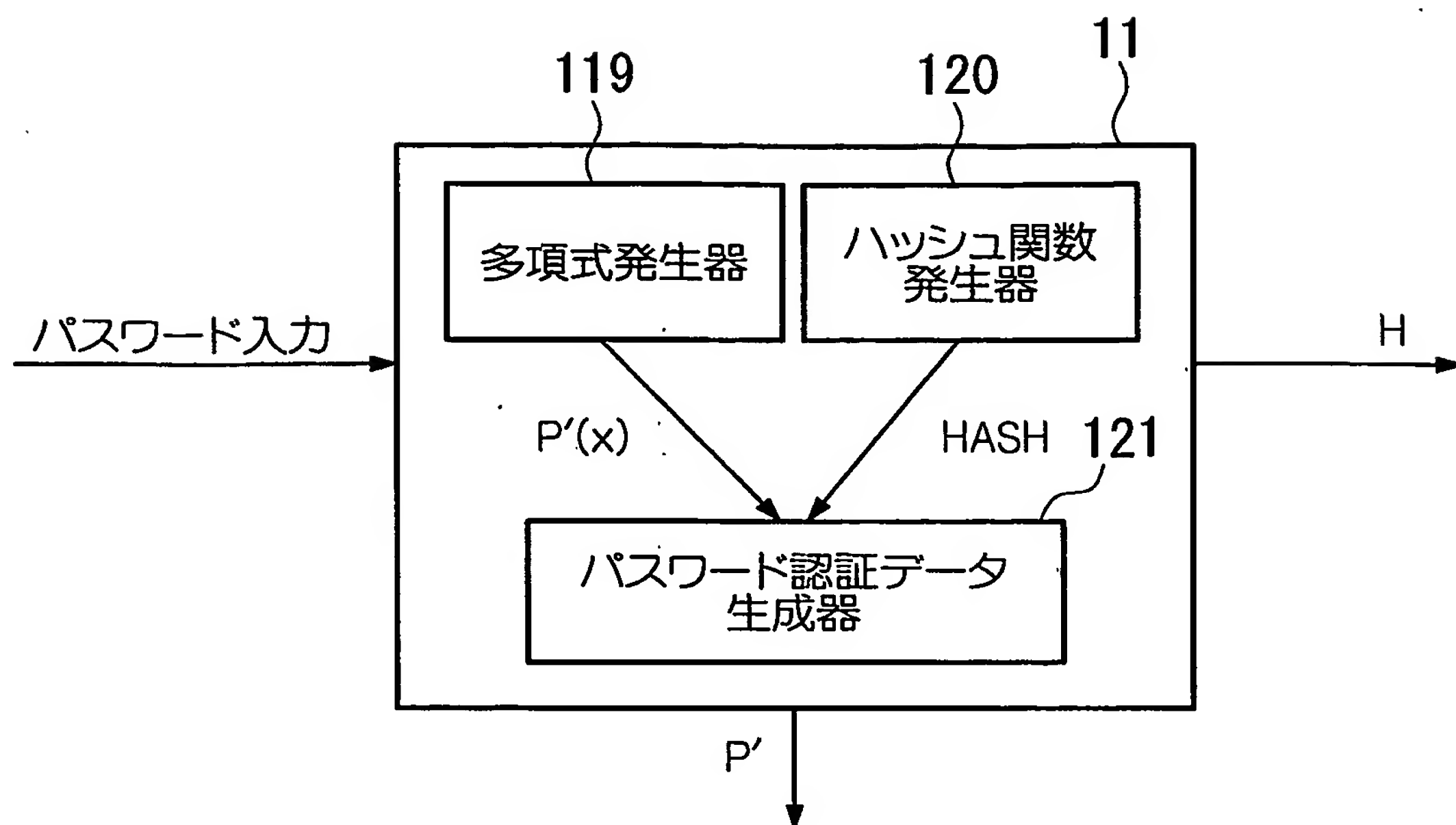
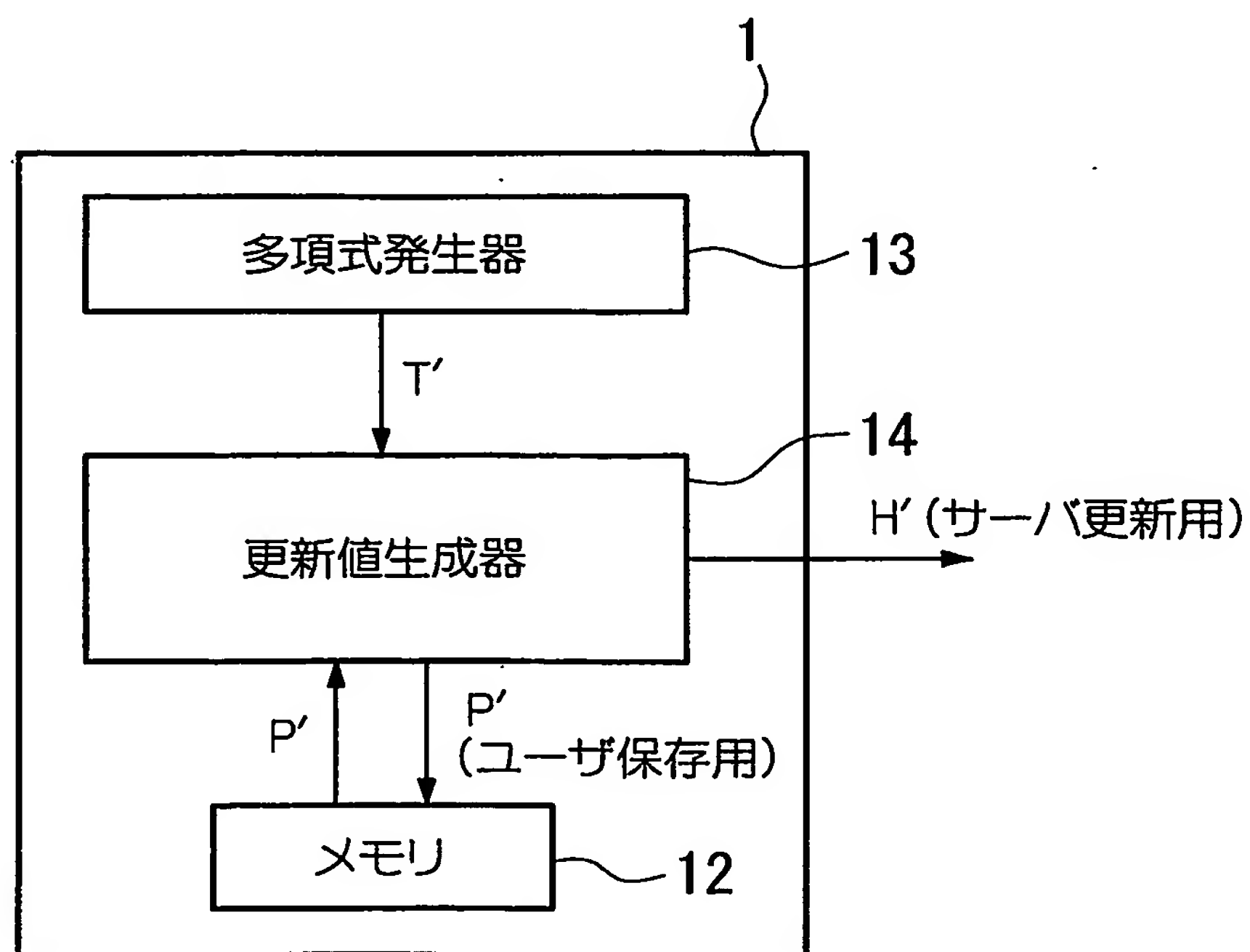


図9



7/22

図10

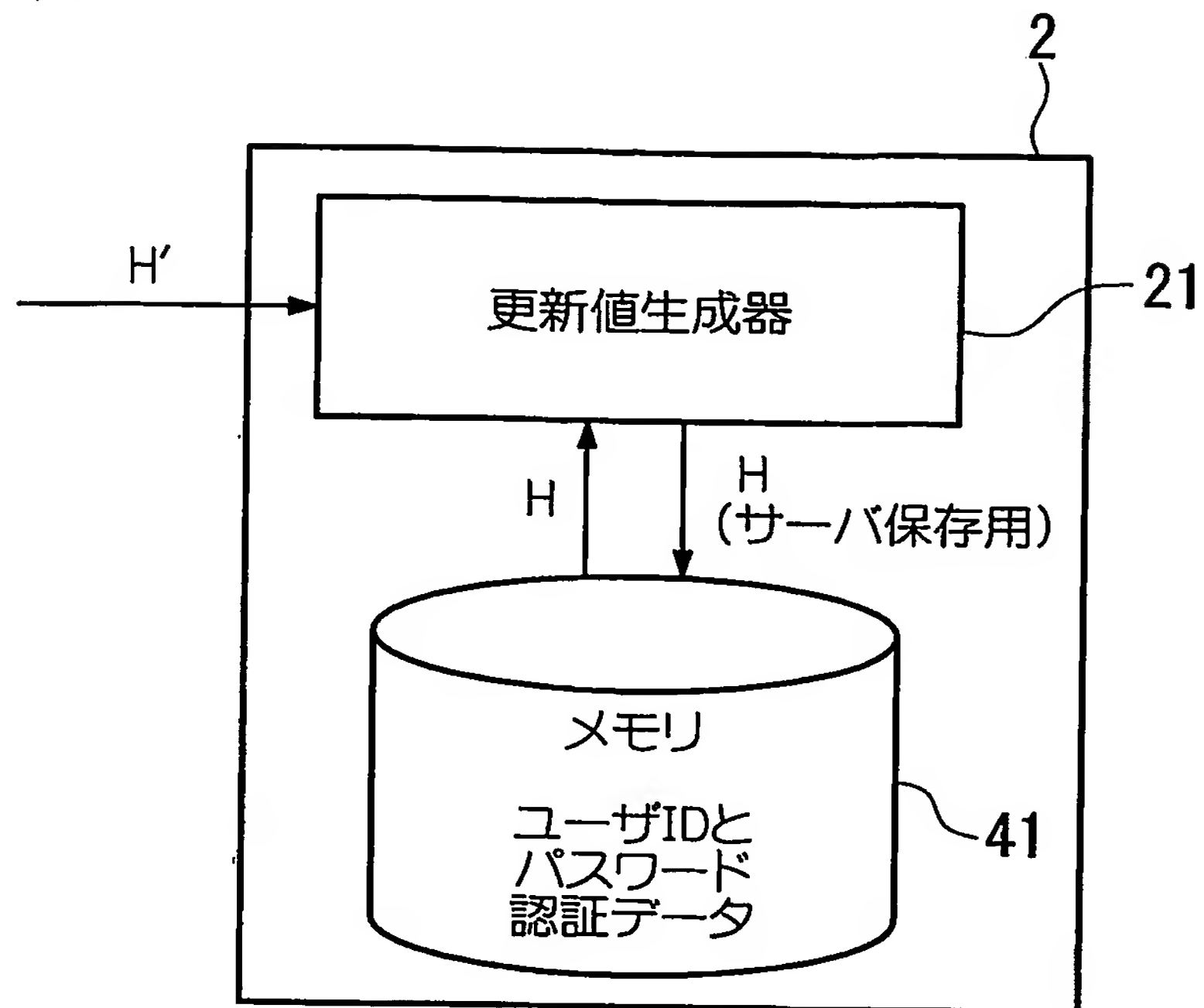
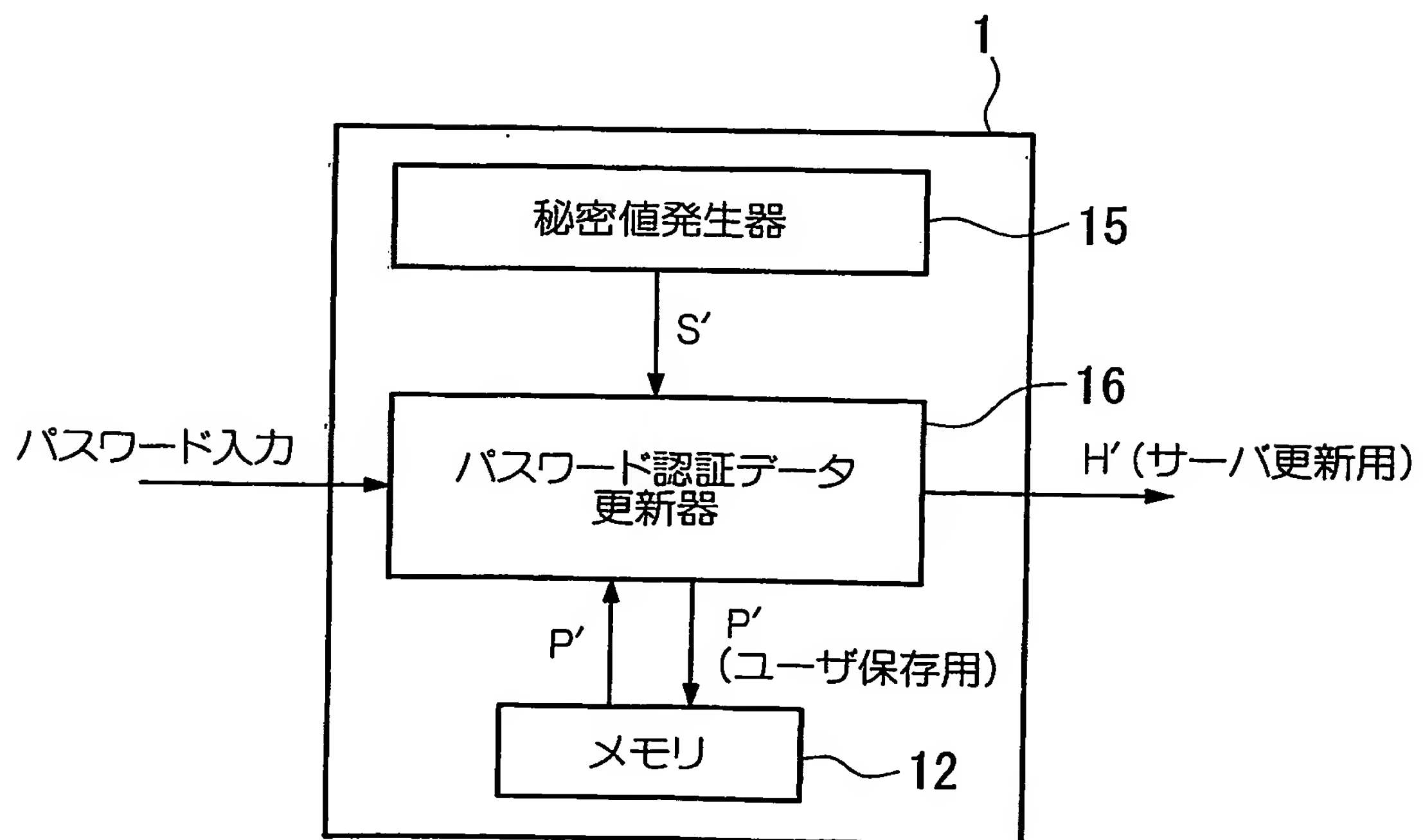


図11



8/22

図12

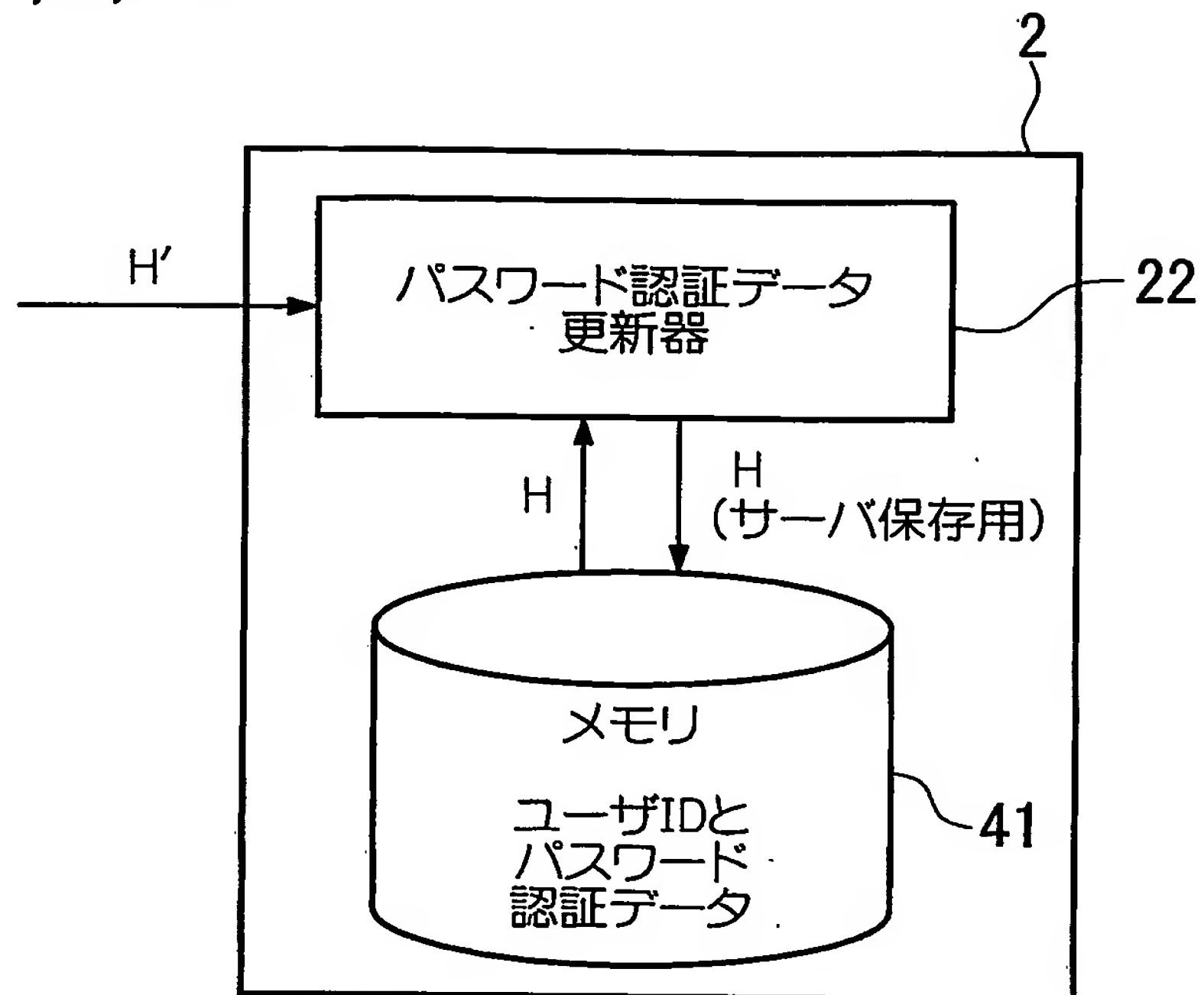
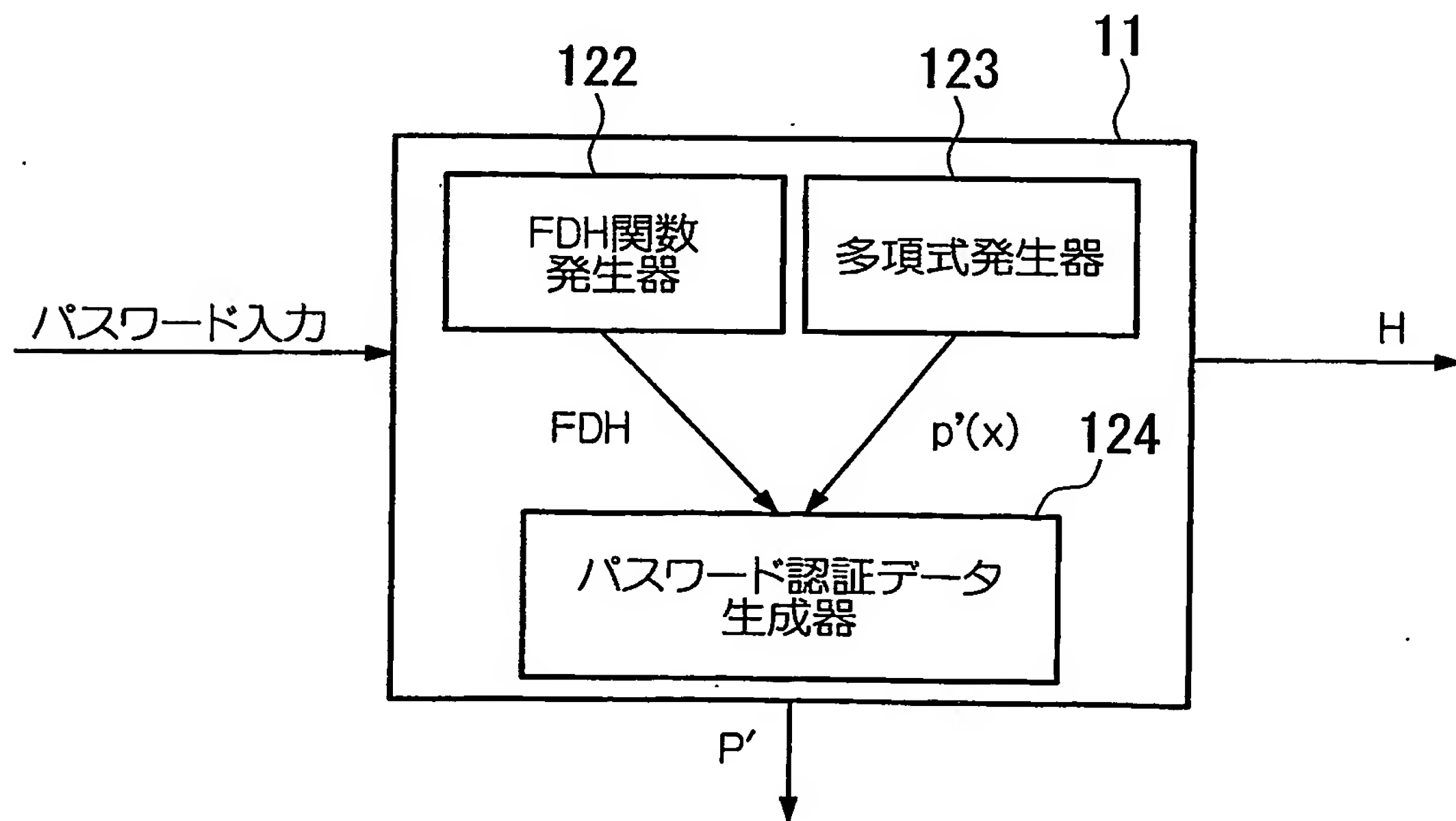


図13



9/22

図14

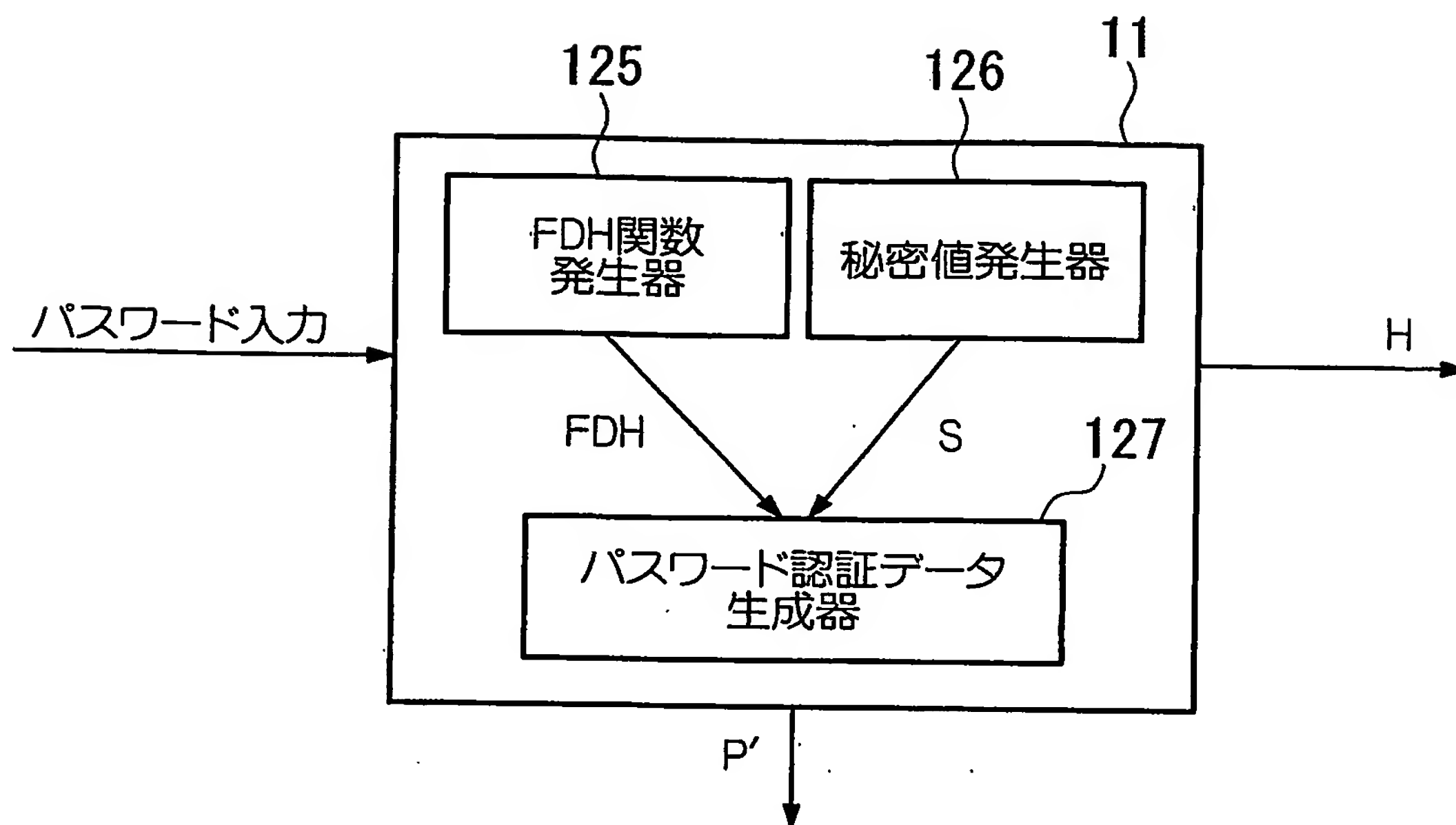
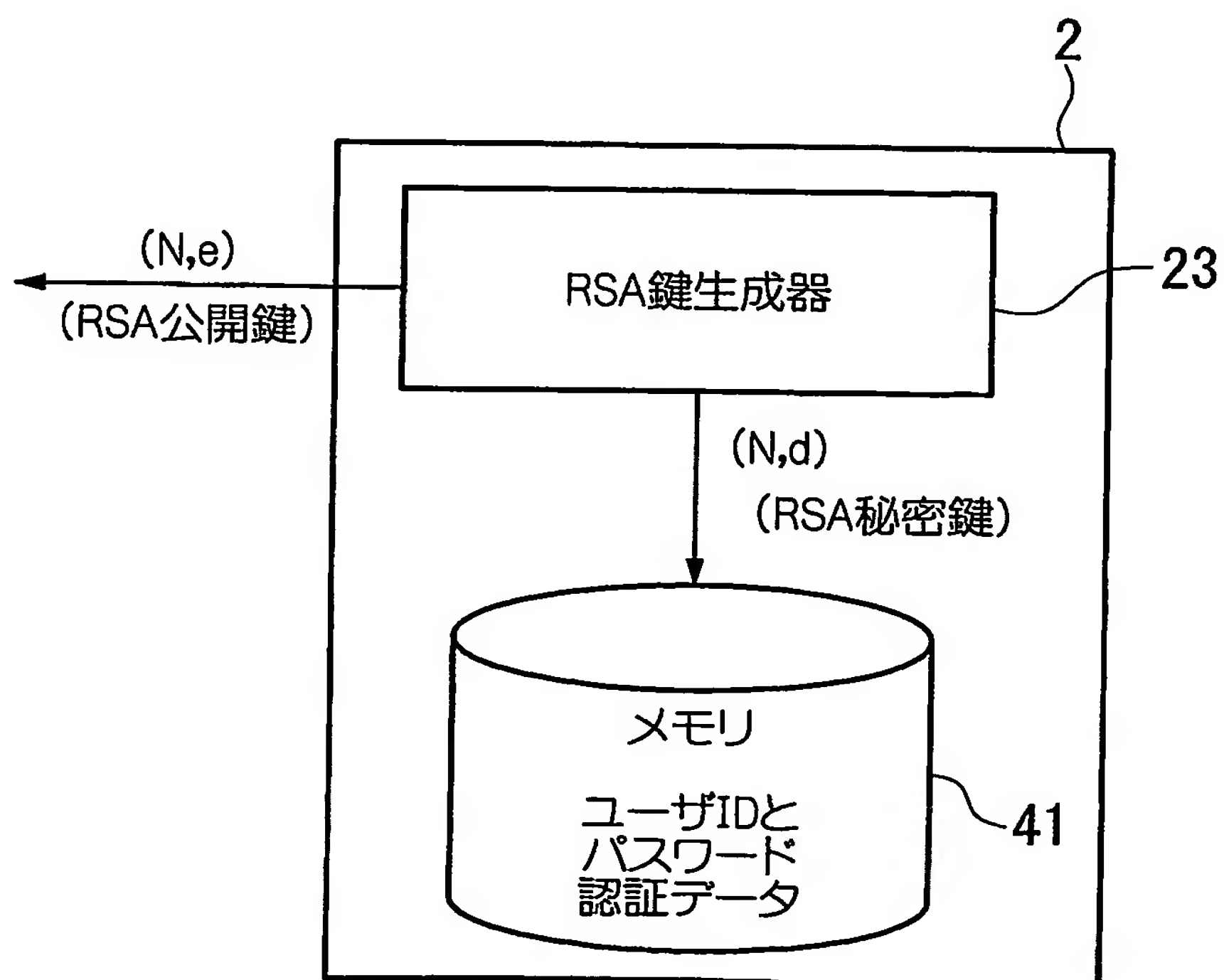
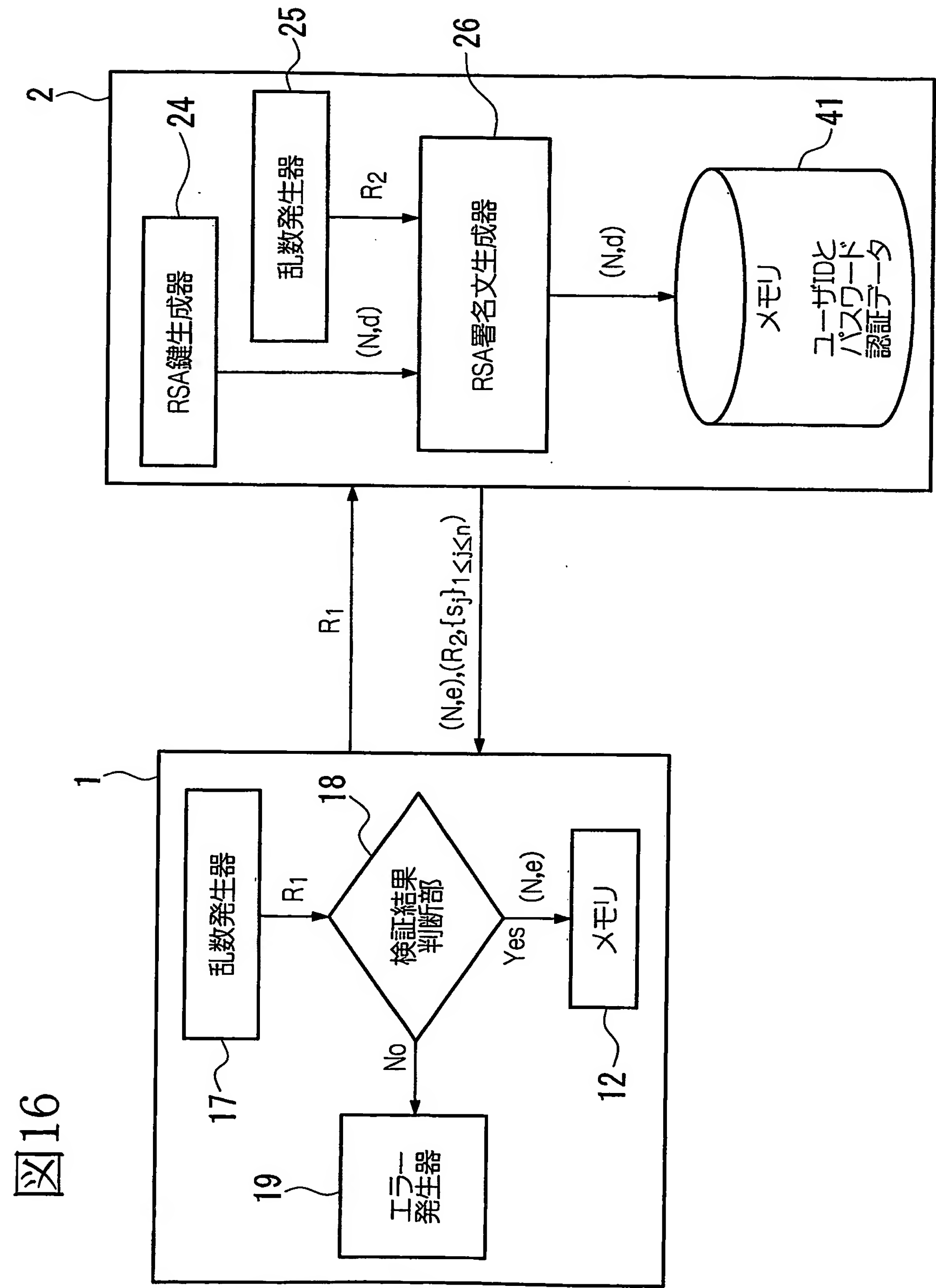
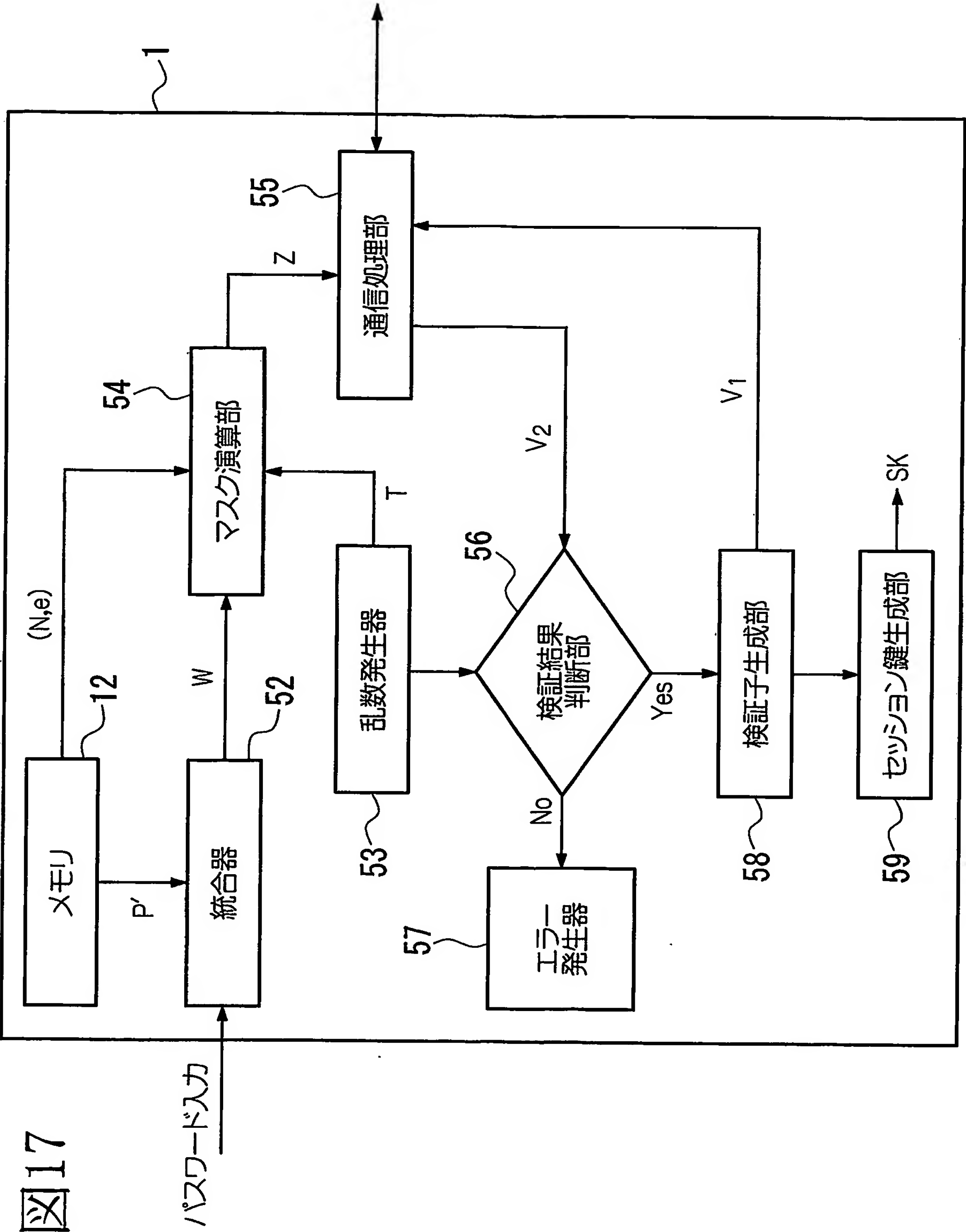


図15







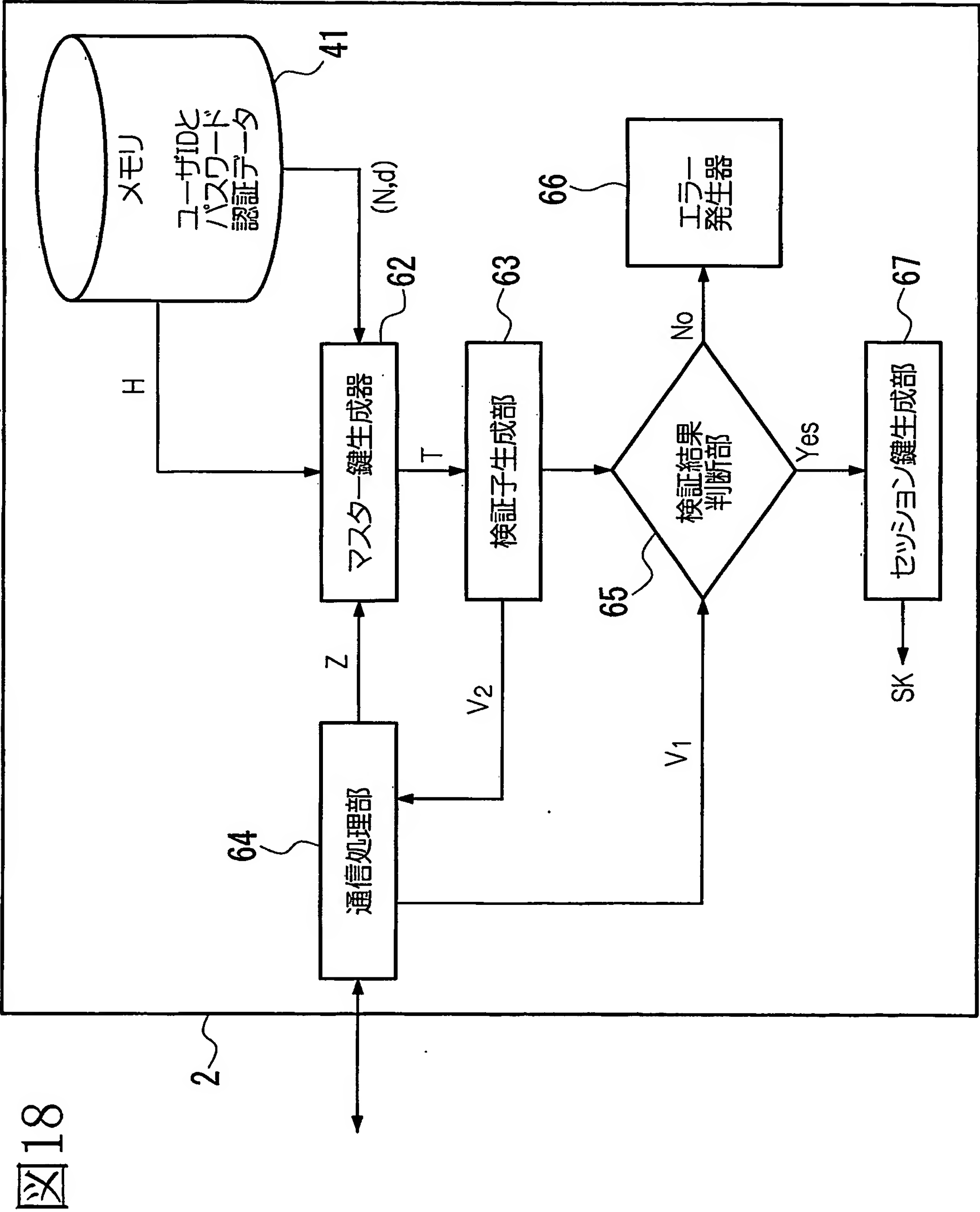
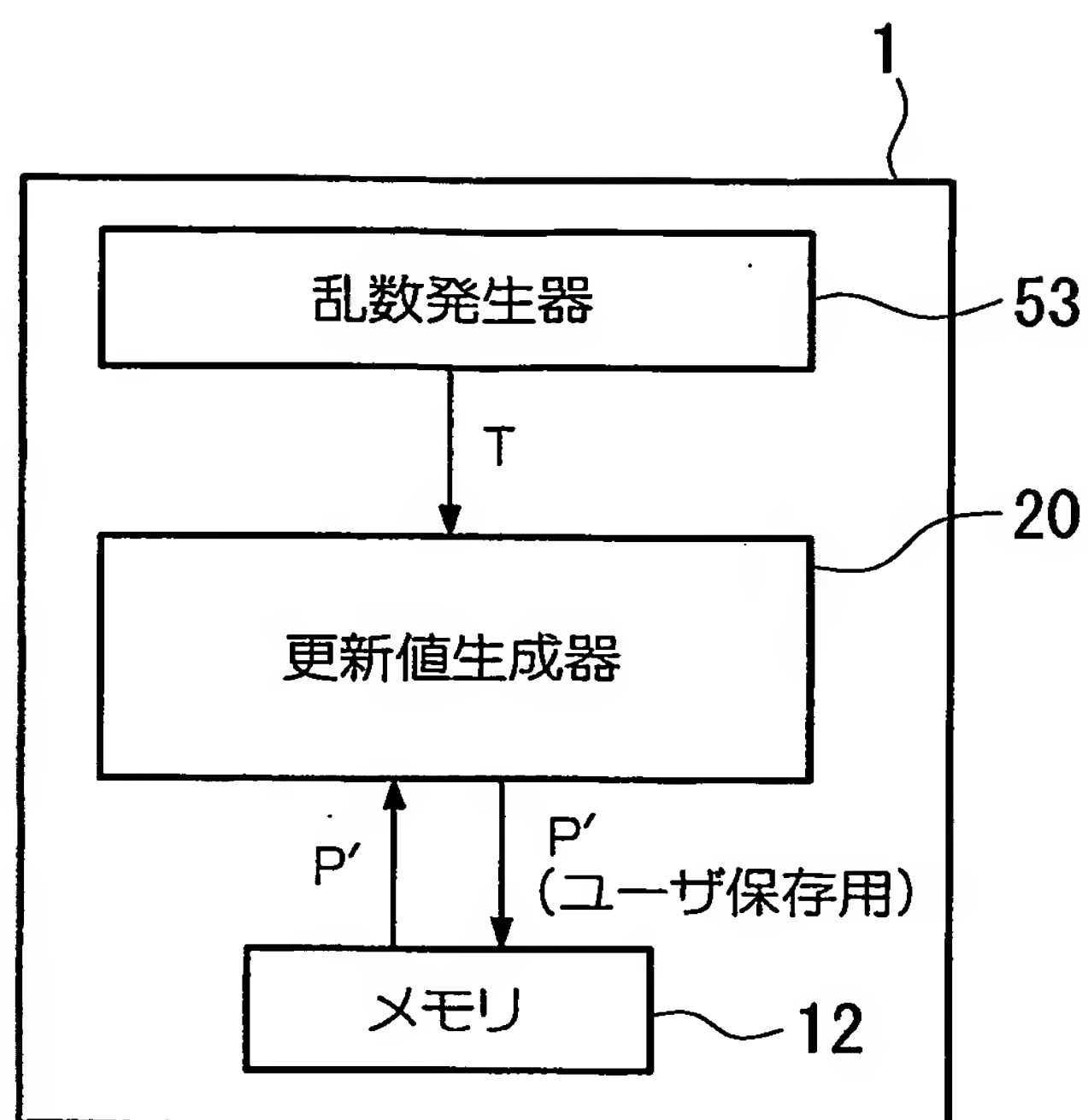


図19



14/22

図20

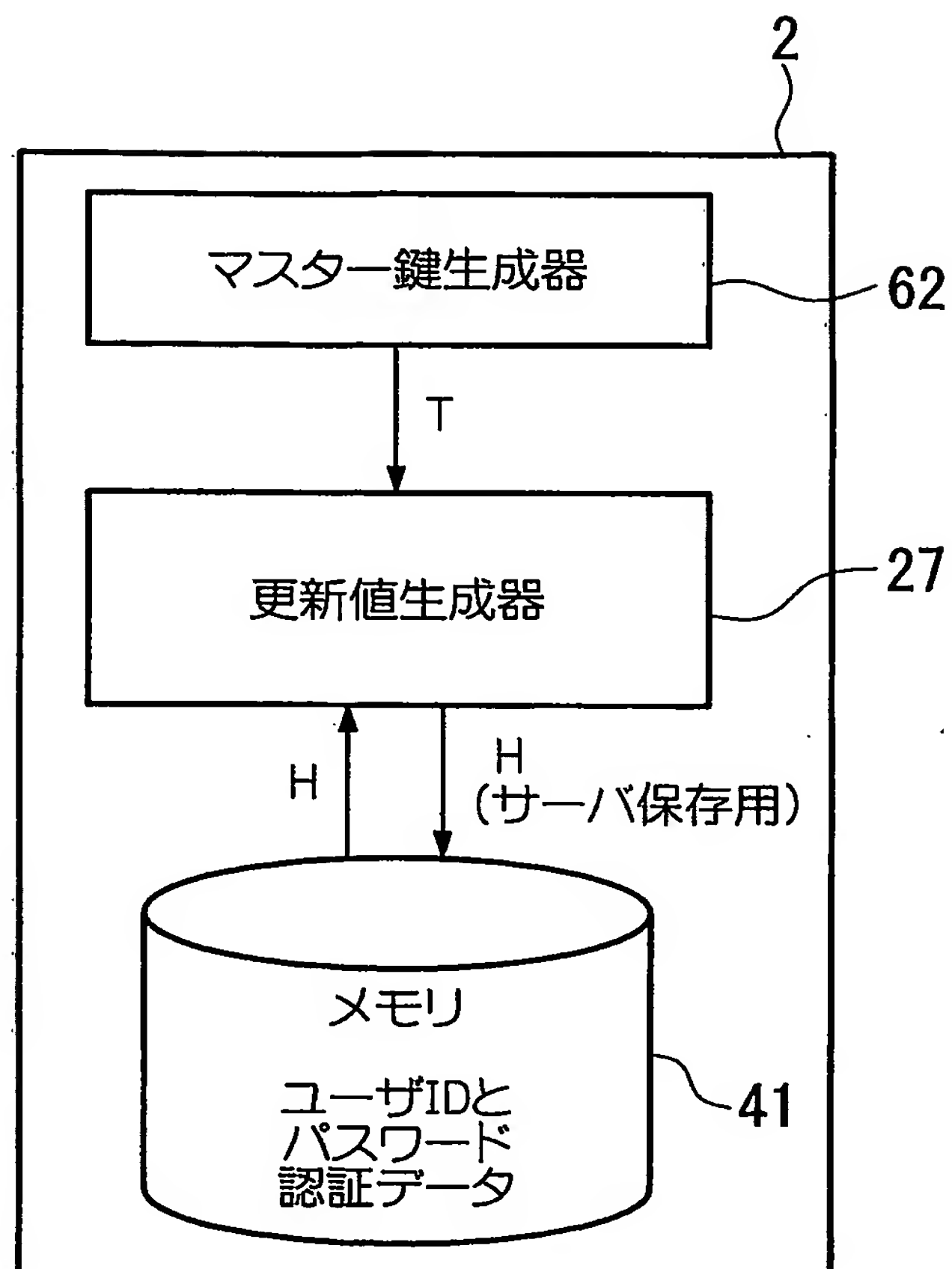


図21

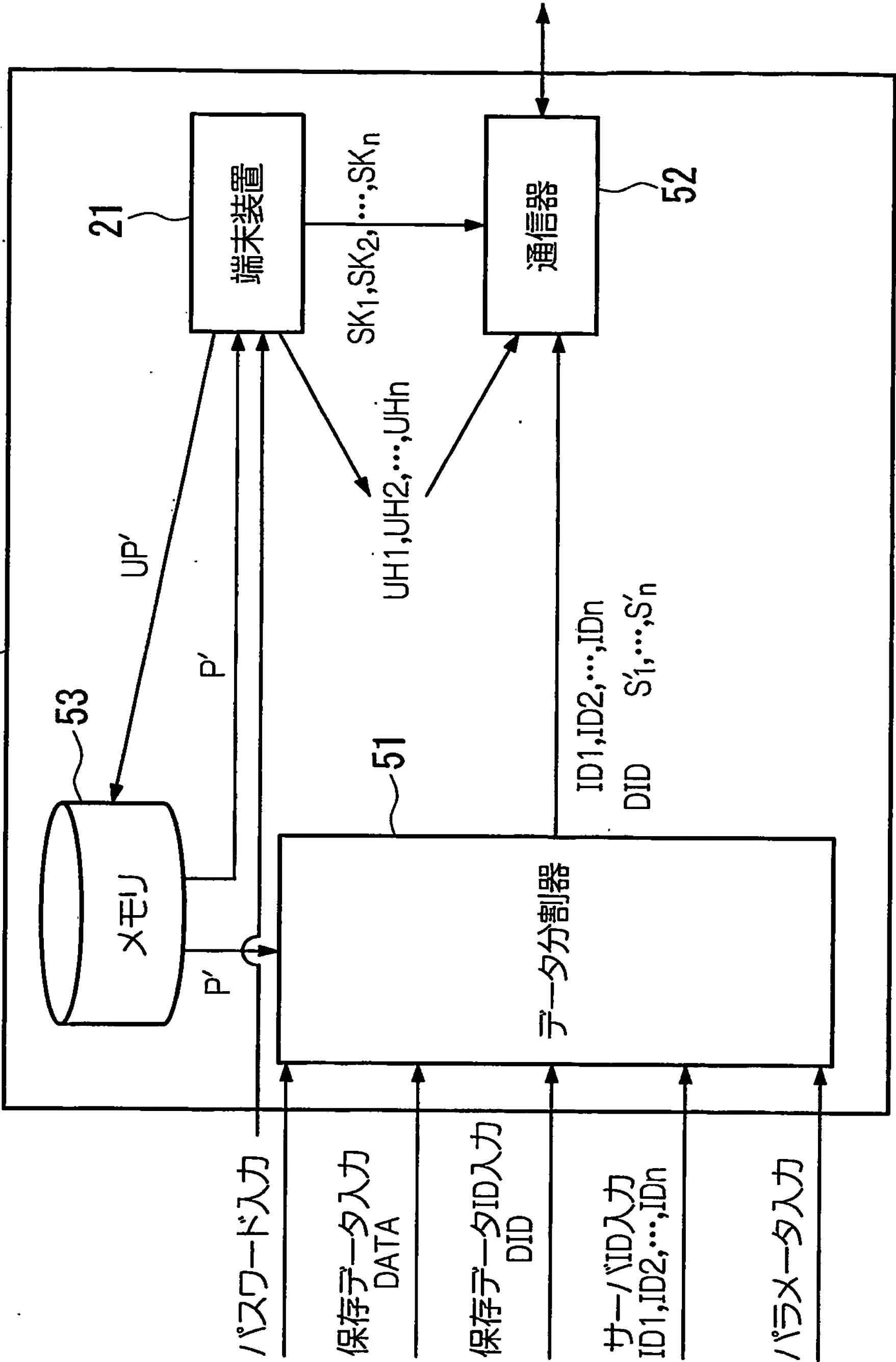


図22

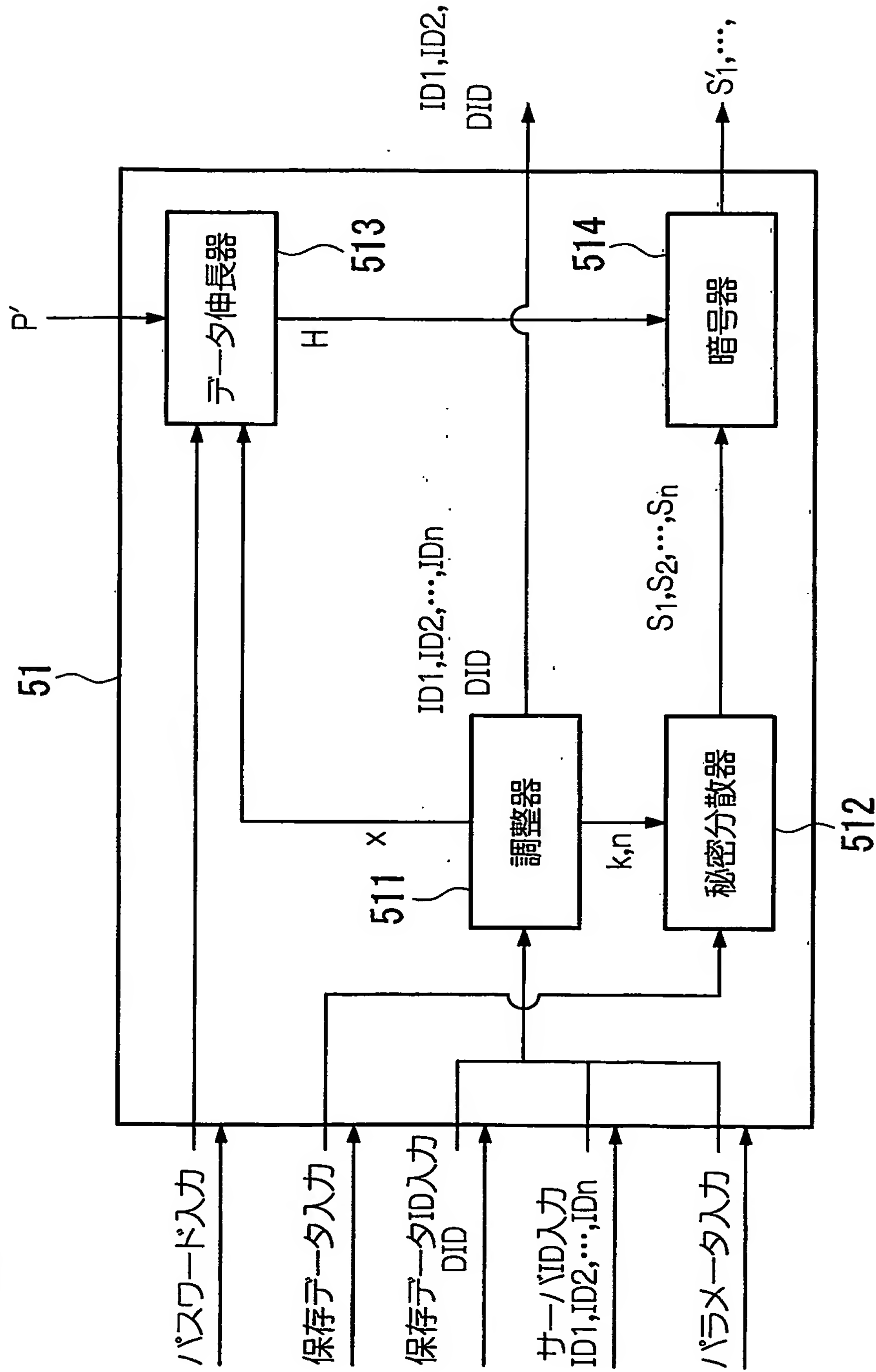


図23

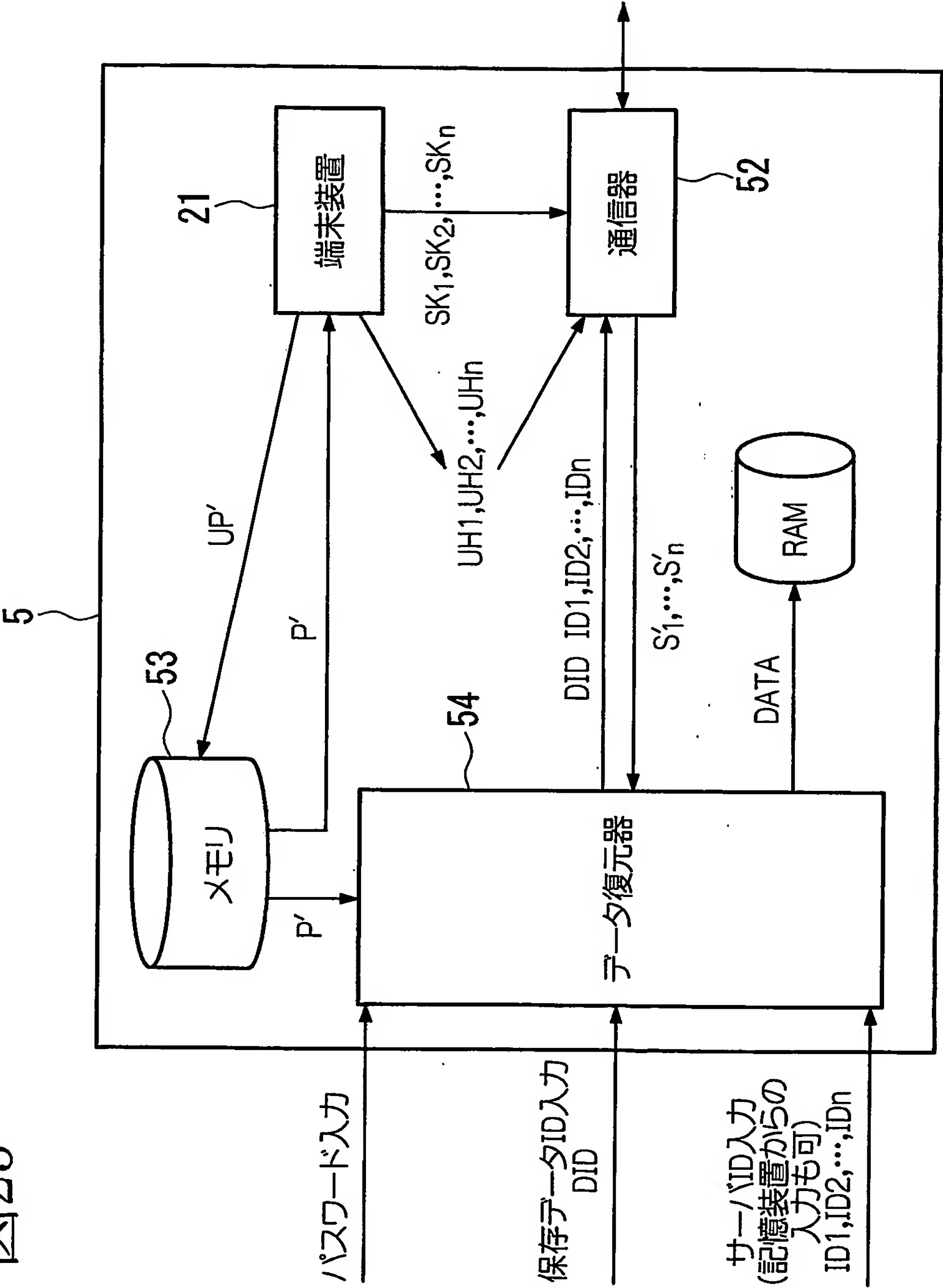


図24

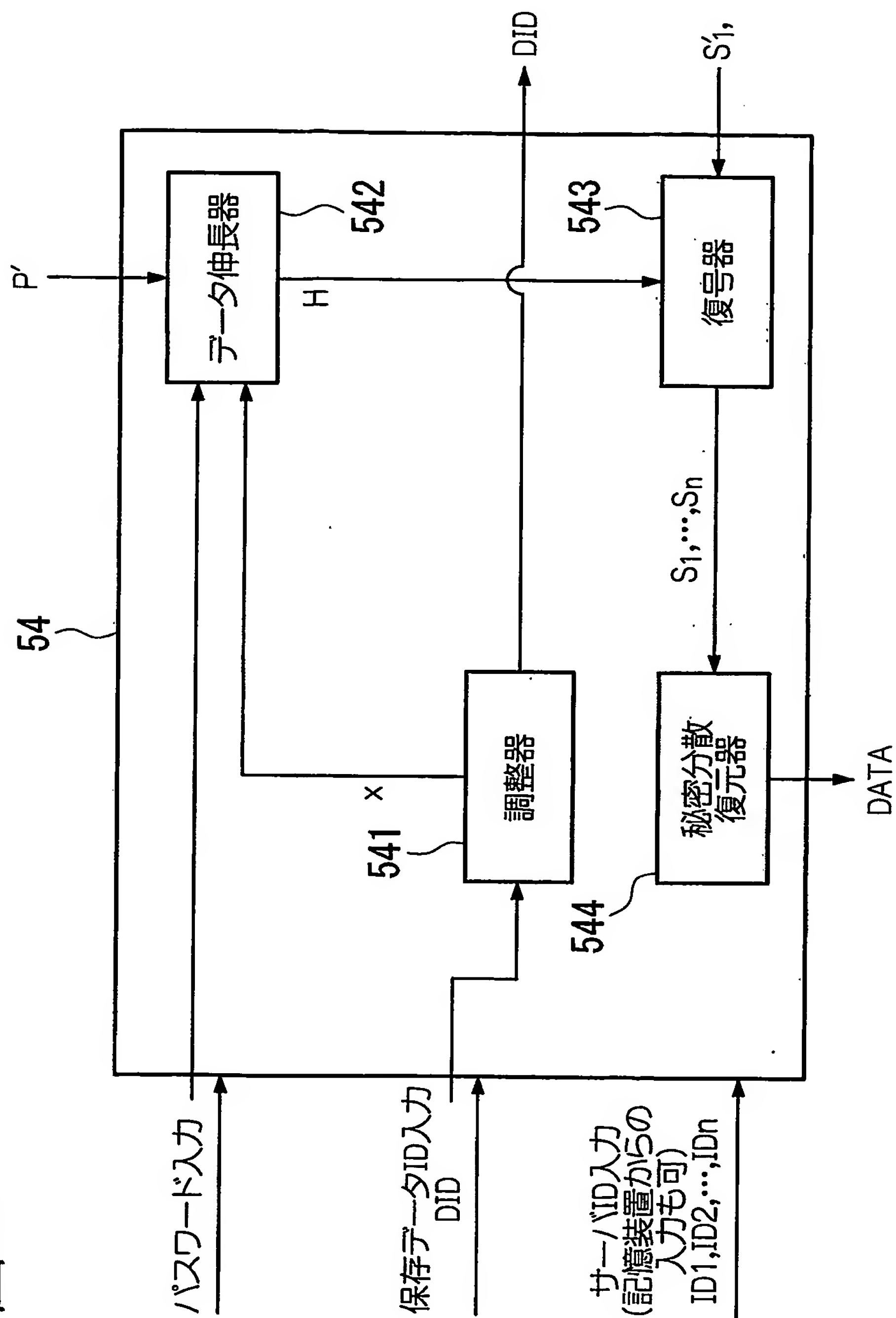


図25

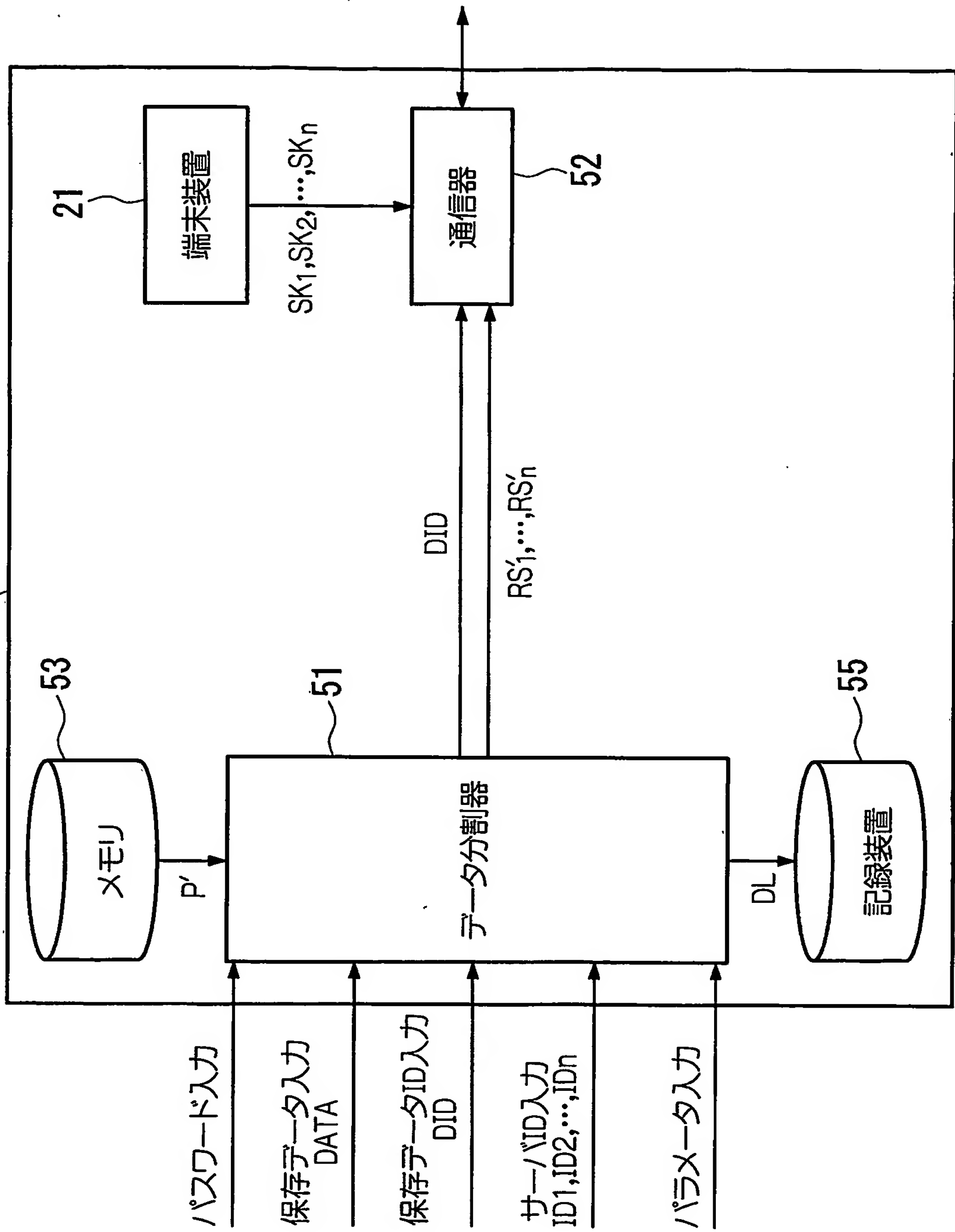
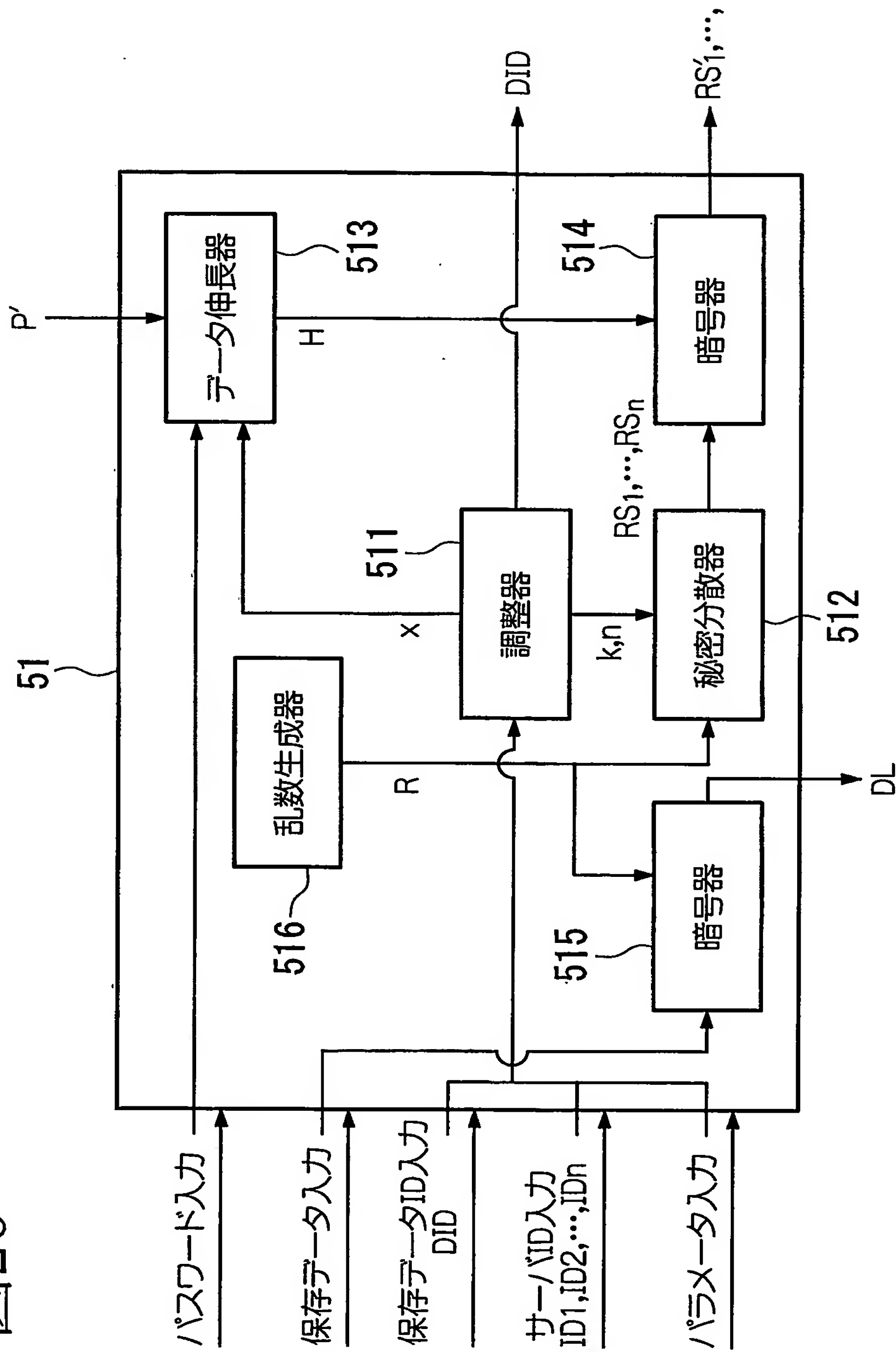


図26



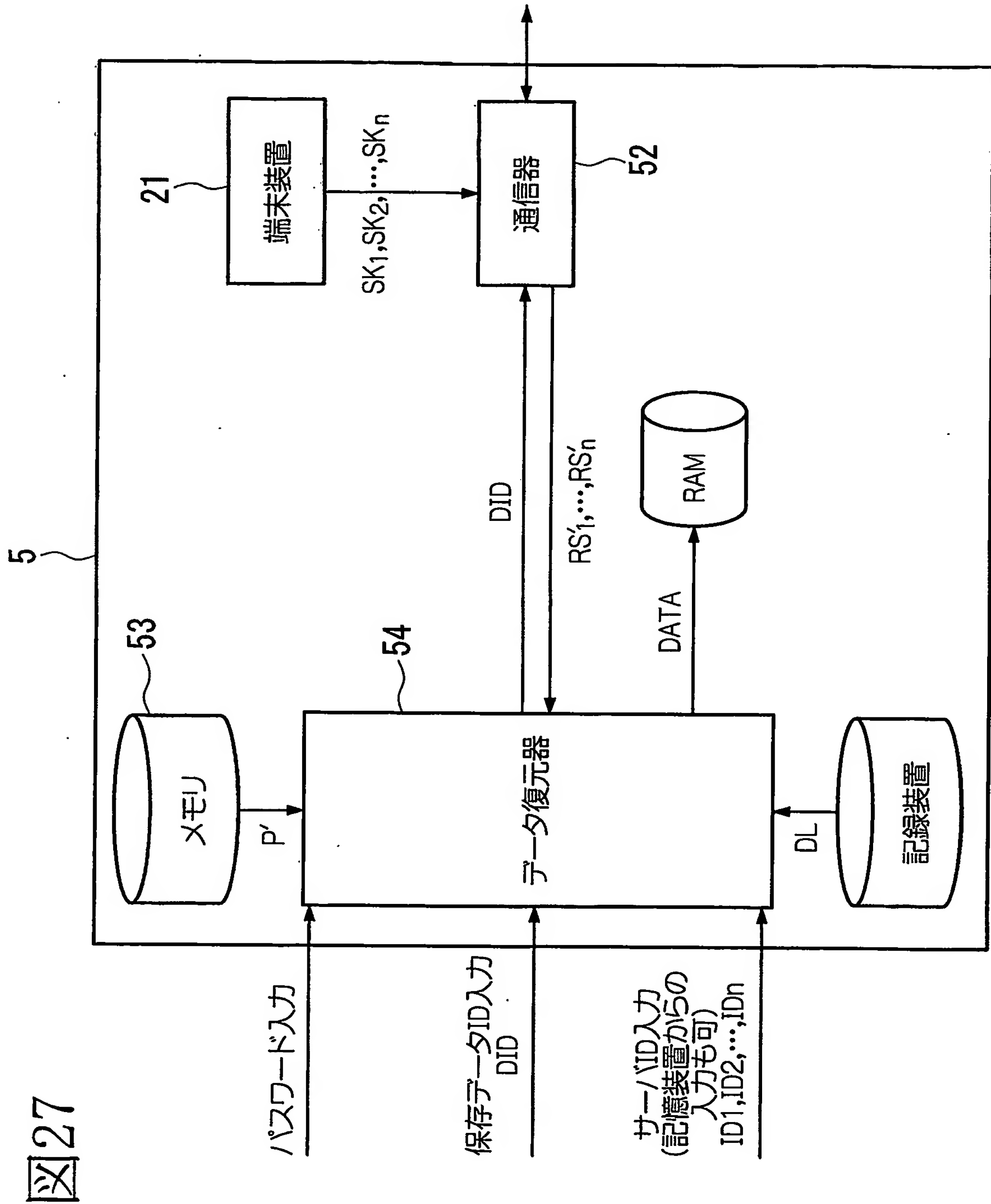
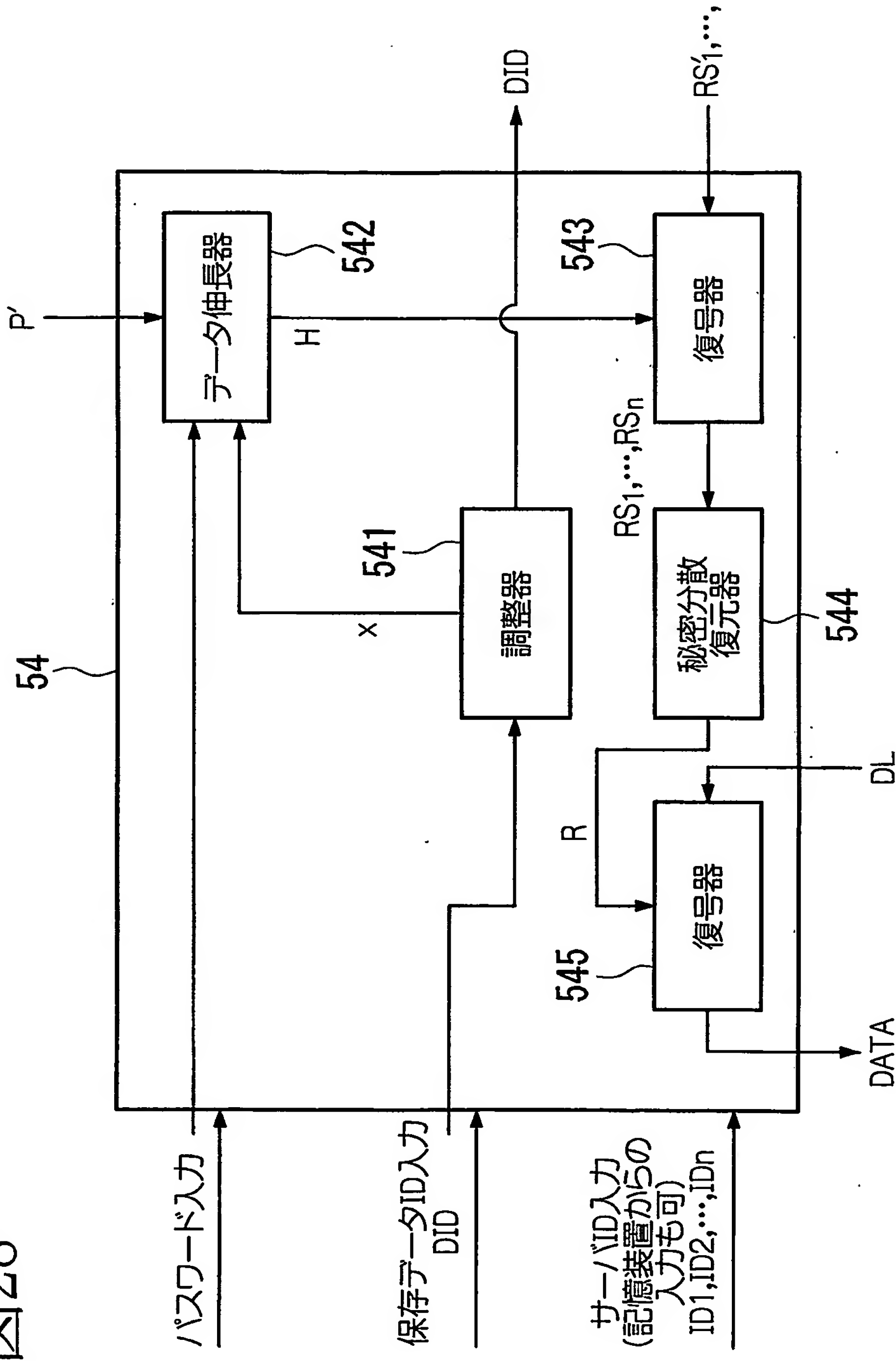


図28



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2004/015184

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl⁷ H04L9/32, H04L9/08, G06F15/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl⁷ H04L9/32, H04L9/08, G06F15/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Toroku Jitsuyo Shinan Koho	1994-2004
Kokai Jitsuyo Shinan Koho	1971-2004	Jitsuyo Shinan Toroku Koho	1996-2004

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

JICST FILE (JOIS)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Kazukuni KOBARA and Hideki IMAI, "Pretty-Simple Password-Authenticated Key-Exchange Protocol Proven to be Secure in the Standard Model", IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E85-A, No.10, 01 October, 2002 (01.10.02), pages 2229 to 2237	1-46
Y	pages 2229 to 2237	47-50
Y	JP 4-245287 A (Matsushita Electric Industrial Co., Ltd.), 01 September, 1992 (01.09.92), Par. No. [0005]; Fig. 1 (Family: none)	47-50

☒ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
10 December, 2004 (10.12.04)

Date of mailing of the international search report
28 December, 2004 (28.12.04)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2004/015184

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P, A	SeongHan Shin, Kazukuni KOBARA, Hideki IMAI, "A New Password-based Authentication Protocol", Computer Security Symposium 2003 (CSS2003), Information Processing Society of Japan, Vol.2003, No.15, 29 October, 2003 (29.10. 03), pages 7 to 12	1-50

A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int. Cl⁷ H04L9/32, H04L9/08, G06F15/00

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int. Cl⁷ H04L9/32, H04L9/08, G06F15/00

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922-1996年
日本国公開実用新案公報	1971-2004年
日本国登録実用新案公報	1994-2004年
日本国実用新案登録公報	1996-2004年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

JICSTファイル (JOIS)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
X Y	Kazukuni KOBARA and Hideki IMAI, “Pretty-Simple Password-Authenticated Key-Exchange Protocol, Proven to be Secure in the Standard Model”, IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, VOL. E85-A, NO. 10, 2002. 10. 01, p. 2229-2237 p. 2229-2237	1-46 47-50

☒ C欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」 特に関連のある文献ではなく、一般的技術水準を示すもの
「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)
「O」 口頭による開示、使用、展示等に言及する文献
「P」 国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献
「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」 同一パテントファミリー文献

国際調査を完了した日

10. 12. 2004

国際調査報告の発送日

28.12.2004

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JP)
郵便番号100-8915
東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

青木 重徳

5M

4.229

電話番号 03-3581-1101 内線 3597

様式PCT/ISA/210 (第2ページの続き) (2004年1月)

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☒ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.